Geometric Algebra
New Foundations, New Insights

SIGGRAPH 2001 Course

Organizers: Ambjorn Naeve and Alyn Rockwood

Description

Geometric Algebra has developed in the last decades from earlier pioneering mathematics of Grassmann and Clifford. It promises to stimulate new methods and insights in all areas of science and engineering dealing with spatial relationships, including computer graphics and related fields. This course first introduces and motivates the topic, and then provides example applications of interest for computer graphics professionals and researchers.

Geometric Algebra unifies many different and redundant mathematical systems in current use. It is especially useful for handling geometric problems, since it allows for intrinsic, i.e., coordinate free, and dimensionally seamless descriptions of geometry. It has wide application in computer graphics, e.g., kinematics and dynamics, simplicial calculations (polygons, FEM), fluid flow, collision detection, quaternion splines , elastic deformations, curve and surface definition, vector fields etc. In all cases, new insights and improved algorithms invariably result. Geometric Algebra is a new and fundamental language for the mathematics of computer graphics, as well as for modeling and interactive techniques in general.

Speakers

Chris Doran, Dept. of Physics, Cambridge University, UK
Leo Dorst, Dept. of Computer Science, University of Amsterdam, the Netherlands
David Hestenes, Dept. of Physics, Arizona State University, Arizona
Joan Lasenby, Dept. of Engineering, Cambridge University, UK
Stephen Mann, Dept. of Computer Science, Waterloo University, Ontario
Ambjorn Naeve, Computing Science, Royal Institute of Technology, Sweden
Alyn Rockwood, Mitsubishi Electric Research Lab, Boston MA.

Table of Contents / Schedule

Biographies

Chris Doran studied at Cambridge University, obtaining a Distinction in Part III Mathematics and a Ph.D. in 1994. He was elected a Junior Research Fellow of Churchill College in 1993, and was made a Lloyd's of London Fellow in 1996. He currently holds an EPSRC Advanced Fellowship, and is a Fellow of Sidney Sussex College, Cambridge. Chris has published widely on aspects of mathematical physics and is currently researching the applications of geometric methods in Engineering. His interests include geometric algebra, computer vision, robotics, general relativity, and quantum field theory.

Leo Dorst received a Ph.D. 1986 from the Applied Physics Department, Delft University of Technology, The Netherlands. His thesis was on accurate geometrical measurements in discretized images. From 1986 to 1992 he worked as senior research scientist at Philips Laboratories, Briarcliff Manor, NY, USA, focusing on robot path planning and task abstraction in goal-directed systems. From 1992 he has been an assistant professor at the University of Amsterdam, The Netherlands, where his continued interest is on planning and representation in autonomous systems, with an emphasis on reasoning with uncertainty, and the use of Geometric Algebra for geometric representation and computation.

David Hestenes is professor of physics, Arizona State University. His principal work has been the programmatic development of Geometric Algebra as a unified mathematical language for science and engineering. Researchers in leading institutions throughout the world are now applying GA to improve software in commercial and government space programs, to biomechanics/robotics and computer vision, to scientific visualization, to conformations of complex molecules, to relativistic quantum theory and to general relativity (3 books & more than 40 papers). He is a fellow of the American Physical Society and an Overseas Fellow of Churchill College (Cambridge) and a *Foundations of Physics* Honoree.

Joan Lasenby studied mathematics at Cambridge University graduating with first class honours in 1981. She gained a Distinction in Part III mathematics in 1983 and a PhD in Radio Astronomy in 1987. She held a Junior Research Fellowship at Trinity Hall College from 1986-9 and worked for Marconi Research Laboratory from 1989-90. She is currently a Royal Society University Research Fellow in the Signal Processing Group of the Cambridge University Engineering Department and a Fellow of Trinity College. Her research interests include applications of geometric algebra in computer vision and robotics, motion analysis and capture, constrained optimization and structural mechanics.

Stephen Mann is an associate professor of computer science at the University of Waterloo where he teaches computer graphics and splines. He received a Ph.D. and M. S. from the University of Washington and a B.A. from the University of California, Berkeley. He has recently spent half a year at the University of Amsterdam working with Geometric Algebra, where he co-developed GABLE. Current interests include splines, surface pasting, blossoming, triangular interpolants, and geometric algebra.

Ambjorn Naeve received a Ph.D. in Computational Geometry, 1993 from the Royal Institute of Technology (KTH), Stockholm. He has been an early advocate of using projective geometry in computational vision. Since 1984 he works with the Computer Vision and Active Perception research group at KTH, where he has headed the development of a number of computer programs that enhance geometrical understanding in various ways - including PDB. Since 1996 Ambjorn also works as a researcher at the Centre for user-oriented Information technology Design (CID) at KTH, where he is concerned with developing principles for the design of interactive learning environments.

Alyn Rockwood received his Ph.D. in applied mathematics from Cambridge University. He has spent 25 years in industrial and academic research, at SGI he developed the NURBS rendering methods for GL/OpenGL; at Evans and Sutherland he developed the first hardware textured graphics system; at Shape Data Ltd. he developed the first commercial automatic blending methods in CAD/CAM and more recently at Arizona State University he was a faculty member and project co-director for a major research project in brain imaging. He has authored several books and 50 articles on computer graphics, and served as the 1999 SIGGRAPH papers' chair.

Relevant web-sites

that provide hands-on experience and extended resources:

1. http://www.cgl.uwaterloo.ca/~smann/GABLE/
   The hands-on interactive tutorial that will be demonstrated in the course. It requires a copy of MATLAB. It will also be available at the CAL.

2. http://modelingnts.la.asu.edu/GC_R&D.html
   Hestenes' web-site with many of the latest research papers in a wide variety of areas and links to other sites.

3. http://www.mrao.cam.ac.uk/~clifford/
   This is the Cambridge University site.

4. http://www-sigproc.eng.cam.ac.uk/vision
   Mainly computer vision applications (GA and non-GA).

5. http://cid.nada.kth.se/il
   Several interactive mathematical learning environments (including GA).

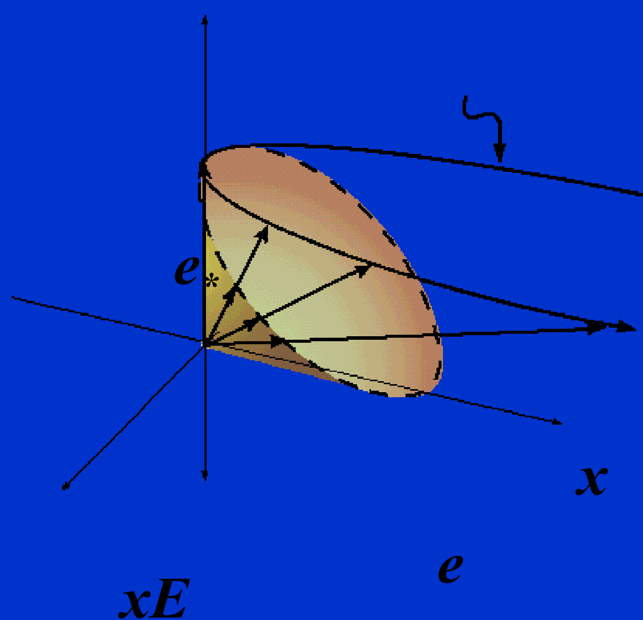
Introduction to Geometric Algebra

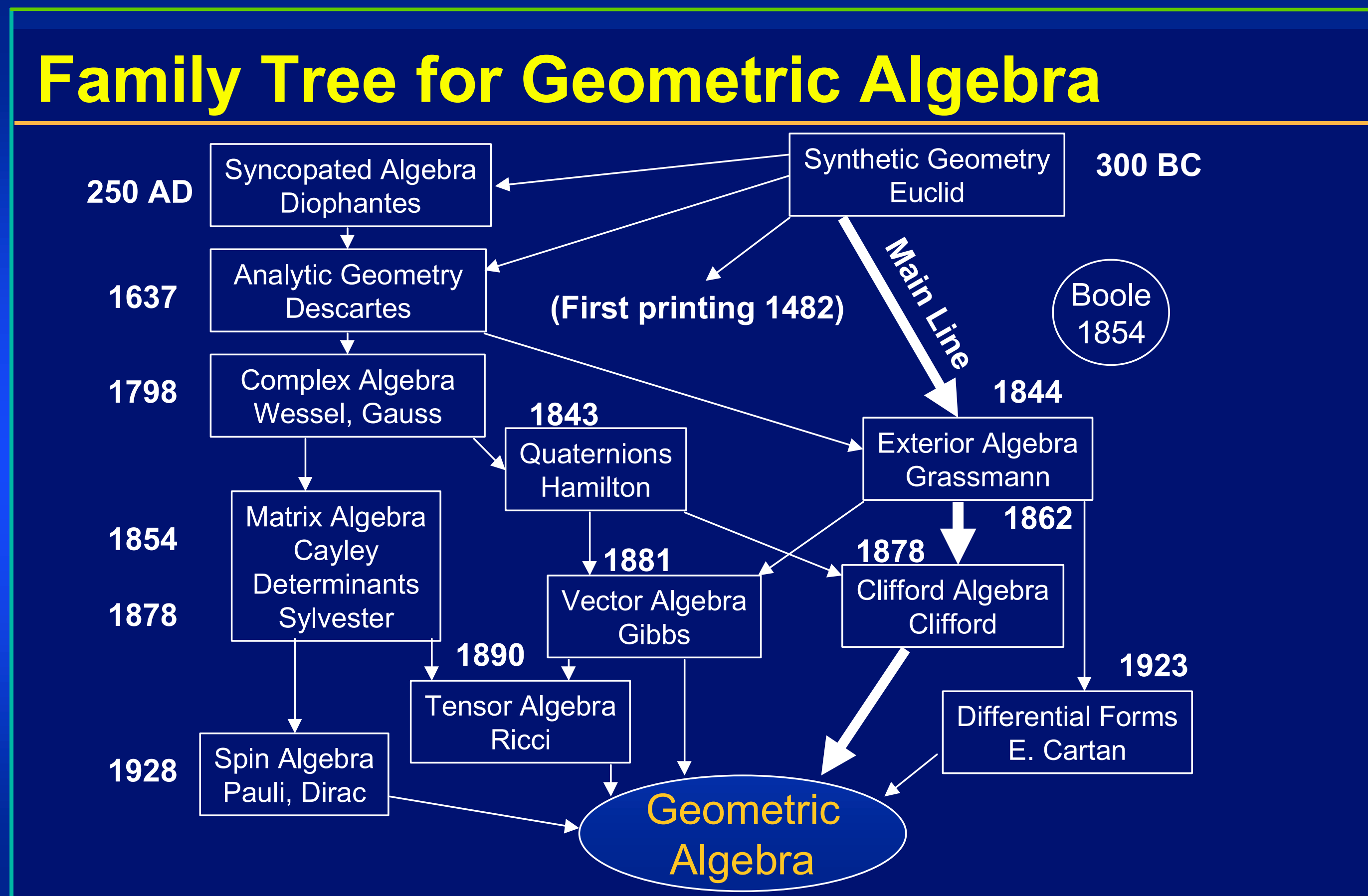
Course # 53

Organizers: Ambjörn Naeve

Alyn Rockwood

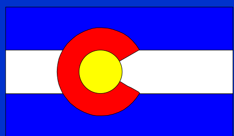
metriframe $(x + e + {}^1/_2 x^2 e)E$



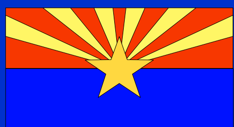


Alyn Rockwood, Oct 1999

Course Speakers



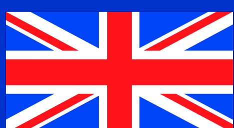
Alyn Rockwood (Colorado School of Mines)



David Hestenes (Arizona State University)

Leo Dorst (University of Amsterdam)

Stephen Mann (University of Waterloo)



Joan Lasenby (Cambridge University)

Chris Doran (Cambridge University)

Ambjörn Naeve (Royal Institute of Technology, Sweden)

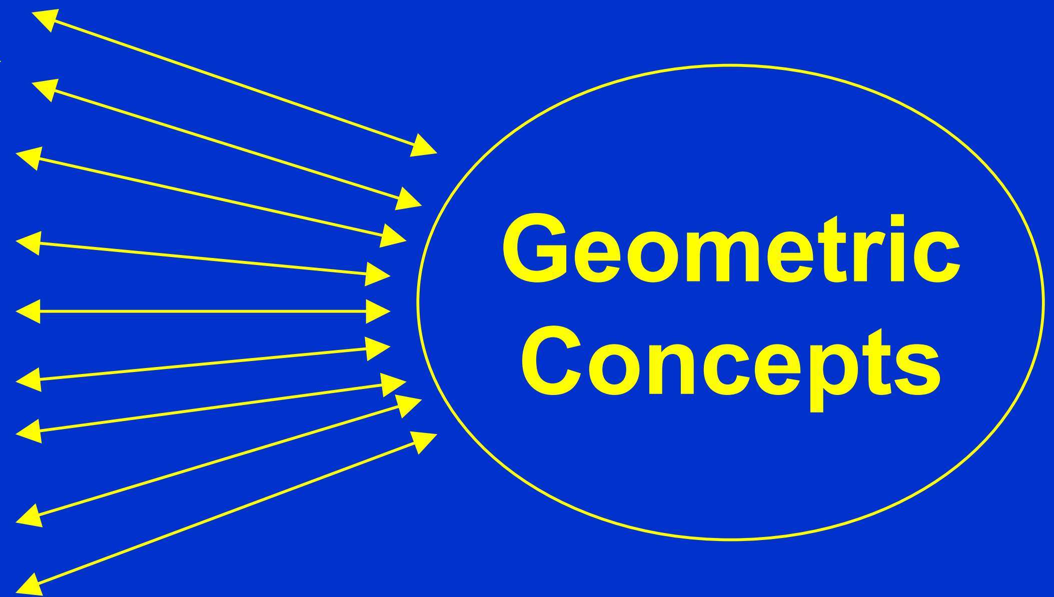
[a] mathematician is a Platonist on weekdays and a Formalist on Sundays. That is, when doing mathematics he is convinced that he is dealing with objective reality … when challenged to give a philosophical account of this reality, he finds it easiest to pretend that he does not believe in it after all. ~P. Davis

Mathematics is Language

	Primitive	
	Nouns	**Verbs**
Vector Algebra	scalar, vector	scalar, dot & cross products, scalar & vector addition gradient, curl, …
Complex Analysis	real, imaginary	addition, multiplication, conjugation, …
Synthetic Geometry	points, line, circles …	intersection, union, …

A Redundant Language

- Synthetic Geometry
  Coordinate Geometry
  Complex Numbers
  Quaternions
  Vector Analysis
  Tensor Analysis
  Matrix Algebra
  Grassmann Algebra
  Clifford Algebra
  Spinor Algebra
  …



- Consequences
  - Redundant learning
  - Complicates knowledge access
  - Frequent translation
  - Lower concept density, i.e., theorems / definitions

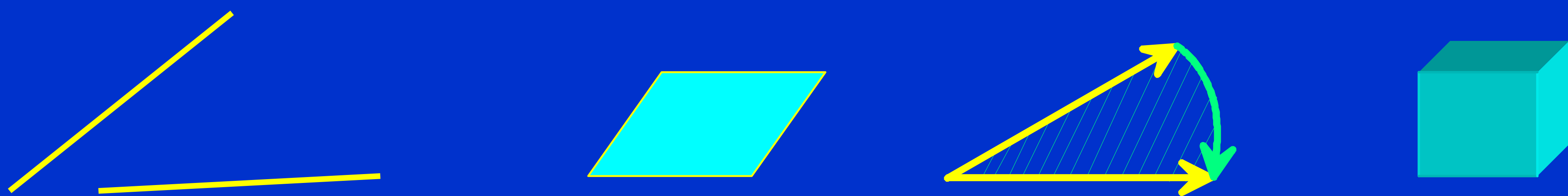
A language for geometry

Hermann Grassmann 1809 - 1977 (Our Hero)

Geometric Concepts → **Algebraic Language**

Properties of nouns

- Grade - dimension
- Direction - orientation, attitude, how it sits in space



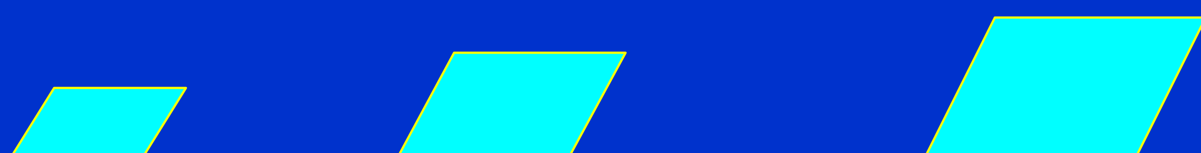
A language for geometry

Hermann Grassmann 1809 - 1977 (Our Hero)

Geometric Concepts → **Algebraic Language**

Properties of nouns

- Grade - dimension
- Direction - orientation, attitude, how it sits in space
- Magnitude - scalar



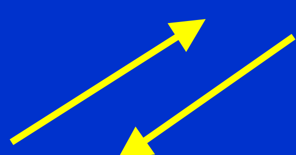
A language for geometry

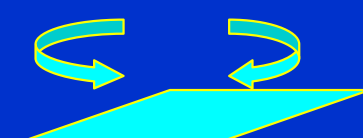
Hermann Grassmann 1809 - 1977 (Our Hero)

Geometric Concepts → **Algebraic Language**

Properties of nouns

- Grade - dimension
- Direction - orientation, attitude, how it sits in space
- Magnitude - scalar
- Sense - positive/negative, up/down, inside/outside





Geometric Algebra

D. Hestenes, *New Foundations for Classical Mechanics*, Kluwer Academic Publishers, 1990

Geometric Concepts → **Algebraic Language**

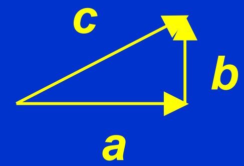
Primitive nouns

• Point	α	scalar	grade 0
• Vector	$\boldsymbol{a}$	directed line	grade 1
• Bivector	$\mathbf{A}$	directed plane	grade 2
• Trivector	$\mathbf{T}$	directed volume	grade 3
• Etc.			

Verbs

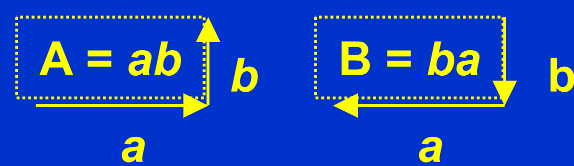
Geometric Concepts → Algebraic Language

- Addition



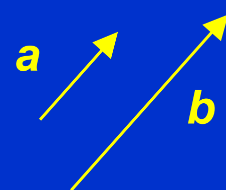
$c = a + b = b + a$

- Multiplication



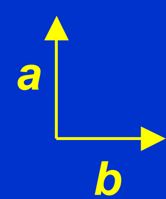
A = -B

- Commutivity



$a \parallel b \Rightarrow ab = ba$

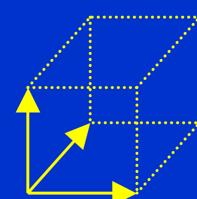
- Anticommutivity



$a \perp b \Rightarrow ab = -ba$

- Associativity

$a \perp b \perp c \Rightarrow (ab)\, c = a\, (bc) = T$



- and others

Prepositions

William Kingston Clifford 1845 - 1879 (Another Hero)

- **Complex analysis**

  Addition defines relation, I.e. $a + i\,b \equiv (a, b)$

- **Clifford's "geometric product" for vectors**

$$ab = a \cdot b \;+\; a \wedge b$$

scalar
(dot product)

bivector
(exterior product)

prepositional add

Geometric Algebra

Nouns *k*-vectors (scalar, vector, bi-vector …)

and multivectors (sums of *k*-vectors)

• Point	α	scalar	grade 0
• Vector	$\boldsymbol{a}$	directed line	grade 1
• Bivector	$\mathbf{A}$	directed plane	grade 2
• Trivector	$\mathbf{T}$	directed volume	grade 3
• …			
• Multivector	$\mathbf{M}$	sum of *k*-vectors	mixed grade

$(M = \alpha + a + A + T + \ldots)$

Geometric Algebra

Verbs Addition

(commutes, associates, identity, inverse)

➡ Familiar operation

Geometric Algebra

Verbs Addition

(commutes, associates, identity, inverse)

Multiplication

(geometric product*)

➡ New operation

Geometric Algebra

Verbs Addition

(commutes, associates, identity, inverse)

Multiplication

(geometric product*)

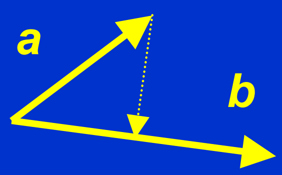
$$\boldsymbol{a}(\alpha + A) = \boldsymbol{a}\alpha + \boldsymbol{a}A$$

Addition and multiplication distribute

The Geometric Product

What can two vectors do?

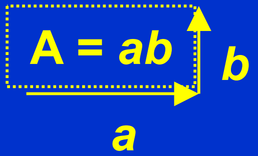
- Project

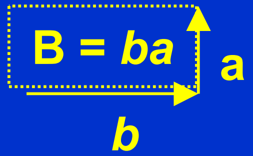


The Geometric Product

What can two vectors do?

- Project
- Define bi-vector



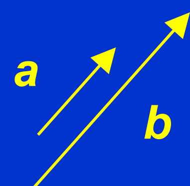


A = -B

The Geometric Product

What can two vectors do?

- Project
- Define bi-vector
- Commute

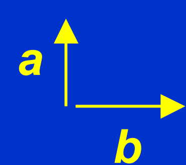


a* || *b* => *ab* = *ba

The Geometric Product

What can two vectors do?

- Project
- Define bi-vector
- Commute
- Anti-commute

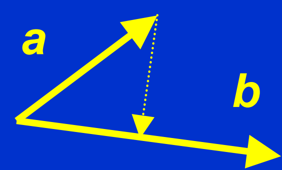


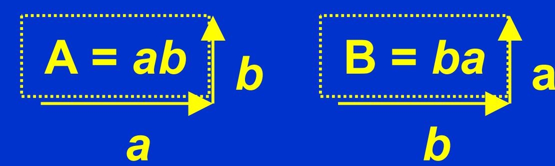
$\boldsymbol{a} \perp \boldsymbol{b} \Rightarrow \boldsymbol{ab} = -\,\boldsymbol{ba}$

The Geometric Product

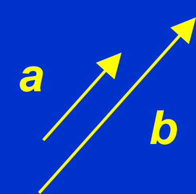
What can two vectors do?

- Project
- Define bi-vector
- Commute
- Anti-commute

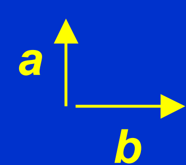




$A = -B$



$a \parallel b \Rightarrow ab = ba$



$a \perp b \Rightarrow ab = -ba$

$$\boxed{ab = a \cdot b + a \wedge b}$$

The Geometric Product

… is more basic!!

Dot product in terms of GP

$$\boldsymbol{a} \cdot \boldsymbol{b} = 1/2\,(\boldsymbol{ab} + \boldsymbol{ba}) \longleftarrow \textbf{scalar}$$

Wedge product in terms of GP

$$\boldsymbol{a} \wedge \boldsymbol{b} = 1/2\,(\boldsymbol{ab} - \boldsymbol{ba}) \longleftarrow \textbf{bivector}$$

Note ⇨ $\boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \wedge \boldsymbol{b} = \boldsymbol{ab}$

Examples

Reflection

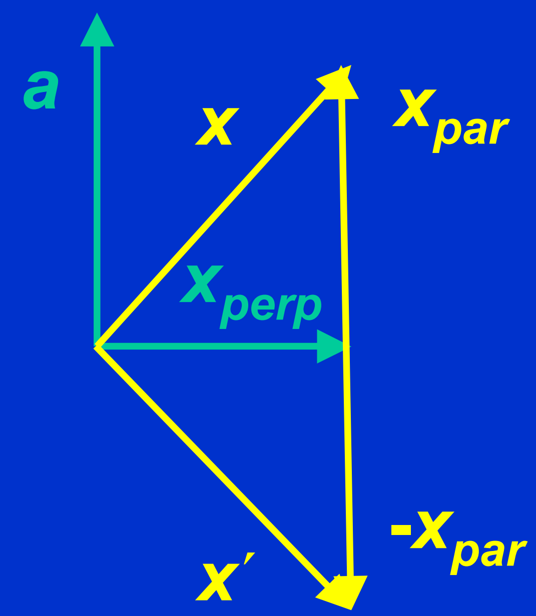
For $a^2 = 1$**,** $-a\,x\,a = -a\,(x_{par} + x_{perp})\,a$

$$= -\,(a\,x_{par} + a\,x_{perp})\,a$$

GA ⟶ $= -\,(x_{par}\,a - x_{perp}\,a)\,a$

$$= -\,(x_{par} - x_{perp})\,a^2$$

$$= -\,x_{par} + x_{perp} = x'$$



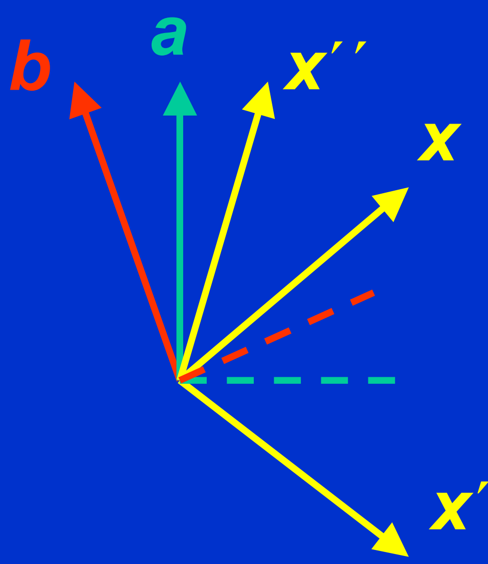
Examples

Rotations ($b^2=1$)

$x'' = -b\, x'\, b = -b\,(-a\, x\, a)\, b = (b\, a)\, x\, (a\, b)$

$(b\, a)\, x\, (a\, b)$ **rotates** x

through 2 $\angle ab$



Examples

Let $a \cdot b = 0$ and $a^2 = b^2 = 1$, define $\boldsymbol{i} = a\,b = -b\,a$

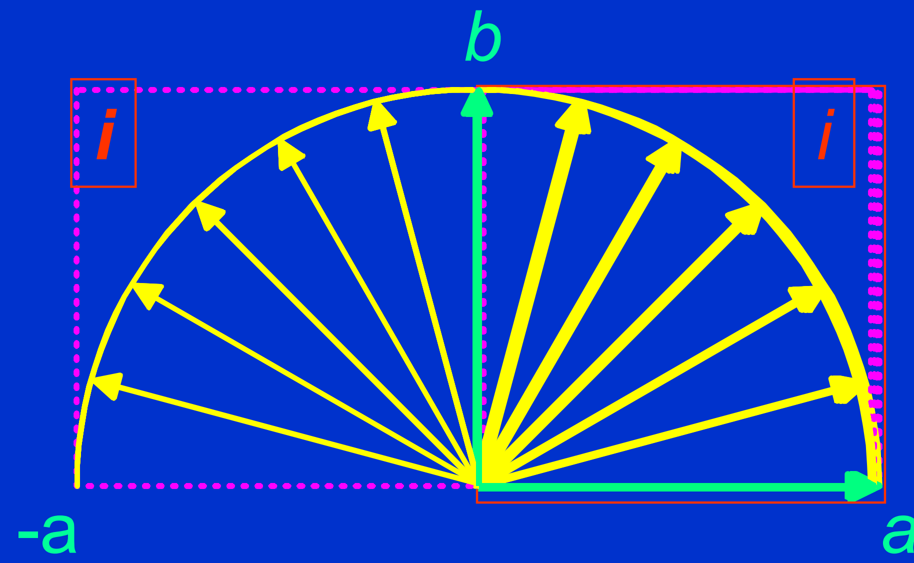
$\boldsymbol{i}$ is an operator:

$a\,\boldsymbol{i} = a\,(a\,b) = a^2\,b = b$

rotates a by 90 degrees to b

$b\,\boldsymbol{i} = (a\,\boldsymbol{i})\,\boldsymbol{i} = a\,\boldsymbol{i}^2 = -a$

rotates a twice, giving $\boldsymbol{i}^2 = -1$

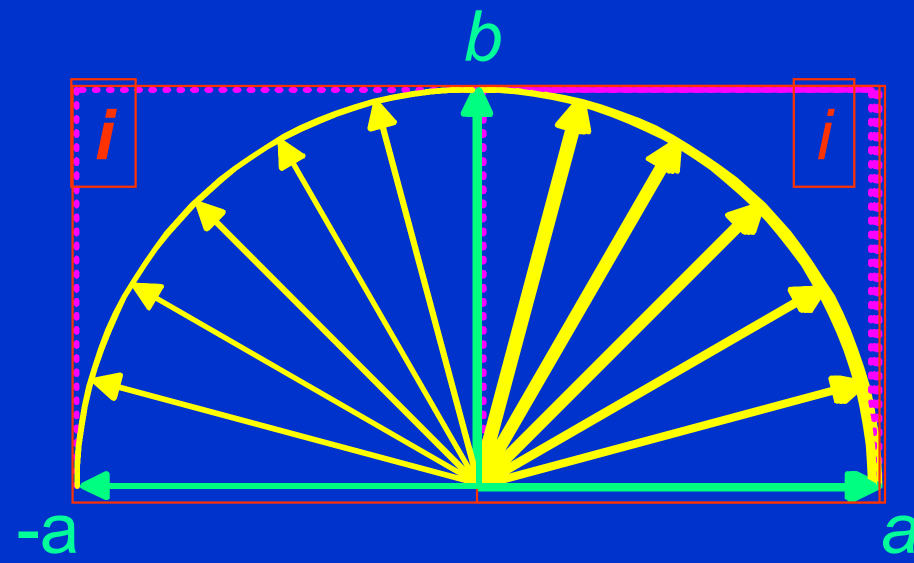


Examples

Let $a \cdot b = 0$ and $a^2 = b^2 = 1$, define $\boldsymbol{i} = a\,b = -b\,a$

$\boldsymbol{i}$ is an operator:

$a\,\boldsymbol{i} = a\,(\,a\,b\,) = a^2\,b = b$
rotates a by 90 degrees to b
$b\,\boldsymbol{i} = (\,a\,\boldsymbol{i}\,)\,\boldsymbol{i} = a\,\boldsymbol{i}^2 = -a$
rotates a twice, giving $\boldsymbol{i}^2 = -1$



Bivectors rotate vectors (!)

Recapitulation

- Graded elements with sense, direction and magnitude
- Addition - verb and preposition
- Geometric product is sum of lower and higher grades
- Dot and Wedge products defined by GP
- Two-sided vector multiplication reflects
- Bivector multiplication rotates vectors
- Special unit bivector ***I*** **(pseudoscalar)**

Axioms

1. **Algebra with non-commutative multiply**

Think of matrix algebra

Axioms

1. Algebra with non-commutative multiply
2. Scalar multiplication commutes $\lambda A = A \lambda$
3. For vector $a^2 = |a|^2 \geq 0$, a scalar
4. $a \cdot A_k$ is a k-1 vector and $a \wedge A_k$ is a k+1 vector

   where $a \cdot A_k = 1/2(aA_k - (1)^{-k} A_k a)$

   and $a \wedge A_k = 1/2(aA_k + (1)^{-k} A_k a)$

Differentiate elements of different grade

Axioms

1. Algebra with non-commutative multiply
2. Scalar multiplication commutes $\lambda A = A \lambda$
3. For vector $a^2 = |a|^2 \geq 0$, a scalar
4. $a \bullet A_k$ is a k-1 vector and $a \wedge A_k$ is a k+1 vector

   where $a \bullet A_k = 1/2(aA_k - (1)^{-k} A_k a)$

   and $a \wedge A_k = 1/2(aA_k + (1)^{-k} A_k a)$

Generalizes dot and wedge products

Axioms

Truncates space to k dimensions

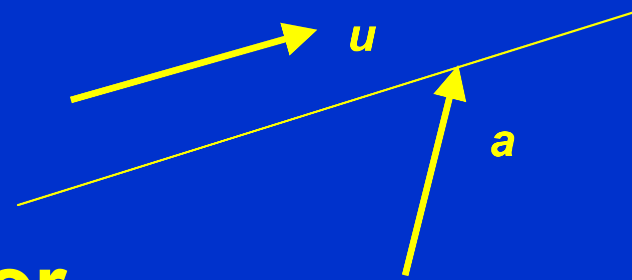
5. $a \wedge A_k = 0$ for a k-dimensional space

Axioms

1. **Non-commutative algebra – add and multiply**
2. **Scalar multiplication commutes $\lambda A = A \lambda$**
3. **For vector $a^2 = |a|^2 \geq 0$, a scalar**
4. **$a \bullet A_k$ is a k-1 vector and $a \wedge A_k$ is a k+1 vector**
   where $a \bullet A_k = 1/2(aA_k - (1)^{-k} A_k a)$
   and $a \wedge A_k = 1/2(aA_k + (1)^{-k} A_k a)$
1. **$a \wedge A_k = 0$ for a k-dimensional space**

Example Algebra

Straight Lines



$(x\text{-}a) \wedge u = 0$ defines line

$\Rightarrow$ $x \wedge u = x \wedge a = \mathrm{M}$, a bivector

$\Rightarrow$ $(x \wedge u)u^{-1} = \mathrm{M}u^{-1}$ (division by vector!)

$\Rightarrow$ $(x \wedge u) \bullet u^{-1} + (x \wedge u) \wedge u^{-1} = \mathrm{M}u^{-1}$ (expansion of GP)

$\Rightarrow$ $(x \wedge u) \bullet u^{-1} + 0 = \mathrm{M}u^{-1}$ (wedging parallel vectors)

$\Rightarrow$ $x - (x \bullet u)\, u^{-1} = \mathrm{M}u^{-1}$ (Laplace reduction theorem)

$\Rightarrow$ $x = (\mathrm{M} + x \bullet u)\, u^{-1}$

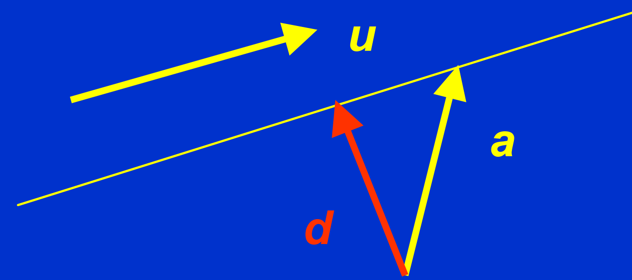
$= (\mathrm{M} + \alpha)\, u^{-1}$

Parametric form for fixed M and *u*.

Example Algebra

Straight Lines

$(x\text{-}a) \wedge u = 0$ defines line



$x = (M + x \bullet u)\, u^{-1}$

$= (M + \alpha)\, u^{-1}$

Parametric form for fixed M and *u*.

…or let $d = M u^{-1}$

$\boxed{x = d + \alpha\, u^{-1}}$, where

$d \bullet u = M u^{-1} \bullet u$

$d \bullet u = 0$ (grade equivalence)

d is orthogonal to *u*

Representations

Complex analysis
Duality
Quaternions
Vector algebra
Spherical geometry

Let $z = ab = a \cdot b + a \wedge b \quad (a^2 = b^2 = 1)$

Let $z^\dagger = ba = (ab)^\dagger \quad$ (reverse = conjugate)

Since $a \cdot b = 1/2(ab + ba) = 1/2(z + z^\dagger)$

$= Re\, z = \lambda\cos\theta$

and $a \wedge b = 1/2(ab - ba) = 1/2(z - z^\dagger)$

$= Im\, z = \lambda i\sin\theta$

then $z = \lambda(\cos\theta + i\sin\theta) = \lambda e^{i\theta}$

The shortest path to truth in the real domain
often passes through the complex domain

Hadamard

Representations

Complex analysis
Duality
Quaternions
Vector algebra
Spherical geometry

Let σ_1, σ_2 and σ_3 be orthonormal basis vectors

then

$$\{1,\ \sigma_1, \sigma_2, \sigma_3,\ \sigma_1\sigma_2, \sigma_1\sigma_3, \sigma_2\sigma_3,\ \sigma_1\sigma_2\sigma_3\}$$

is a basis for the geometric algebra over R^3

Representations

Complex analysis
Duality
Quaternions
Vector algebra
Spherical geometry

Let σ_1, σ_2 and σ_3 be orthonormal basis vectors

then

$$\{1, \quad \sigma_1, \sigma_2, \sigma_3, \quad \sigma_1\sigma_2, \sigma_1\sigma_3, \sigma_2\sigma_3, \quad \sigma_1\sigma_2\sigma_3\}$$

Scalar **vector** **bivector** **trivector**

Let $I = \sigma_1\sigma_2\sigma_3$, the pseudoscalar

What are: $(\sigma_1\sigma_2\sigma_3)^2 = I^2$? $\quad I\sigma_1$? $\quad I\sigma_1\sigma_2$?

Representations

Complex analysis

Duality

Quaternions

Vector algebra

Spherical geometry

Let σ_1, σ_2 and σ_3 be orthonormal basis vectors

then

Let $I = \sigma_1 \sigma_2 \sigma_3$, the pseudoscalar

What are: $(\sigma_1 \sigma_2 \sigma_3)^2 = I^2$? $I \sigma_1$? $I \sigma_1 \sigma_2$?

Representations

Complex analysis

Duality

Quaternions

Vector algebra

Spherical geometry

$$(\sigma_1 \sigma_2 \sigma_3)^2 = I^2 = -1$$

$$I \sigma_1 \text{ ?}$$

$$I \sigma_1 \sigma_2 \text{ ?}$$

Representations

Complex analysis
Duality
Quaternions
Vector algebra
Spherical geometry

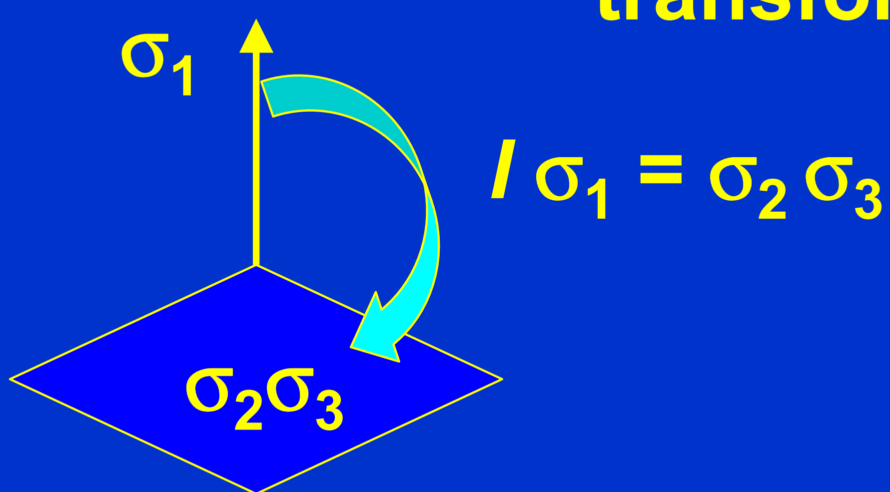
$(\sigma_1 \sigma_2 \sigma_3)^2 = I^2 = -1$

$I \sigma_1 = \sigma_2 \sigma_3$

transforms σ_1 to $\sigma_2\sigma_3$

$I \sigma_1 \sigma_2 = \sigma_3$

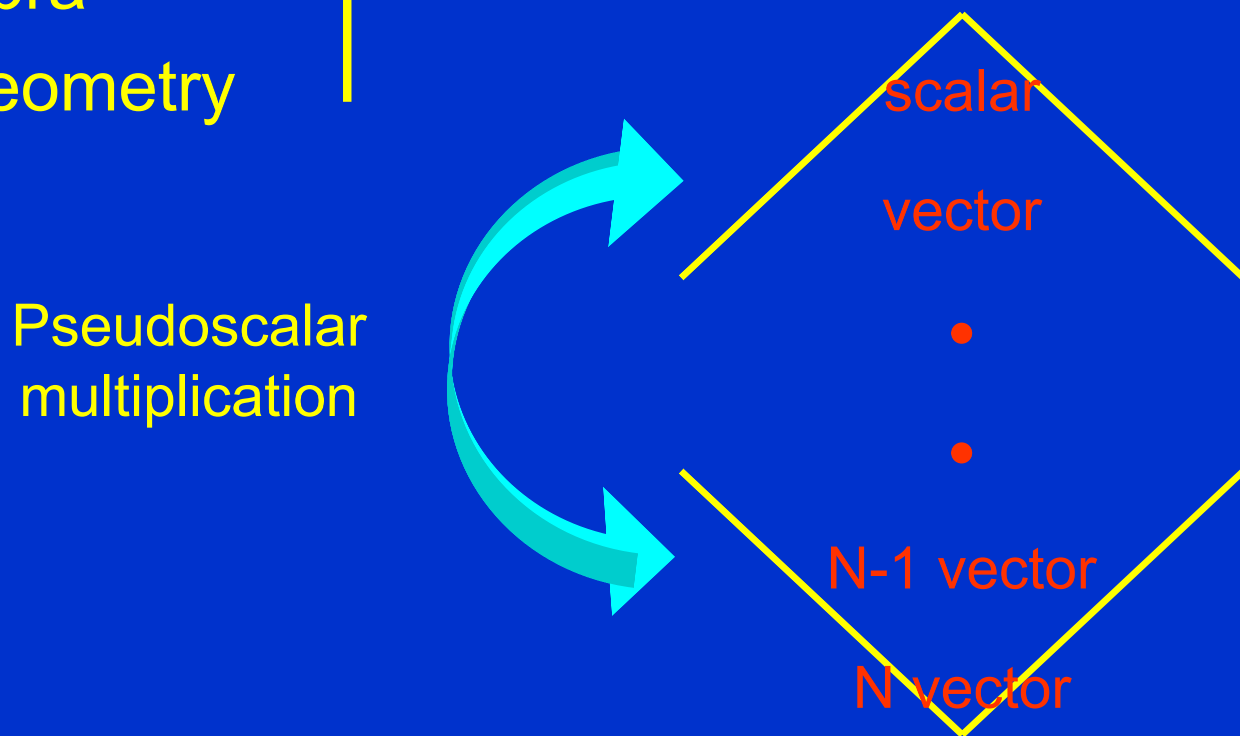
transforms $\sigma_1 \sigma_2$ to σ_3



Representations

- Complex analysis
- Duality
- Quaternions
- Vector algebra
- Spherical geometry

In general:



Representations

Complex analysis
Duality
Quaternions
Vector algebra
Spherical geometry

Let $i = \sigma_2 \sigma_3$

$j = \sigma_3 \sigma_1$

$k = \sigma_1 \sigma_2$

then

$$i^2 = j^2 = k^2 = -1 \quad \text{and} \quad i\,j\,k = -1$$

Hamilton's equations for quaternions!

Representations

Complex analysis
Duality
Quaternions
Vector algebra
Spherical geometry

If (s, q_1, q_2, q_3) is a quaternion

then

$$R = s + i\,q_1 + j\,q_2 + k\,q_3$$

scalar **bivector**

Is a general rotor in GA **Recall $x' = RxR^\dagger$**

Note: *i*, *j*, *k* are bivectors!

Representations

Complex analysis

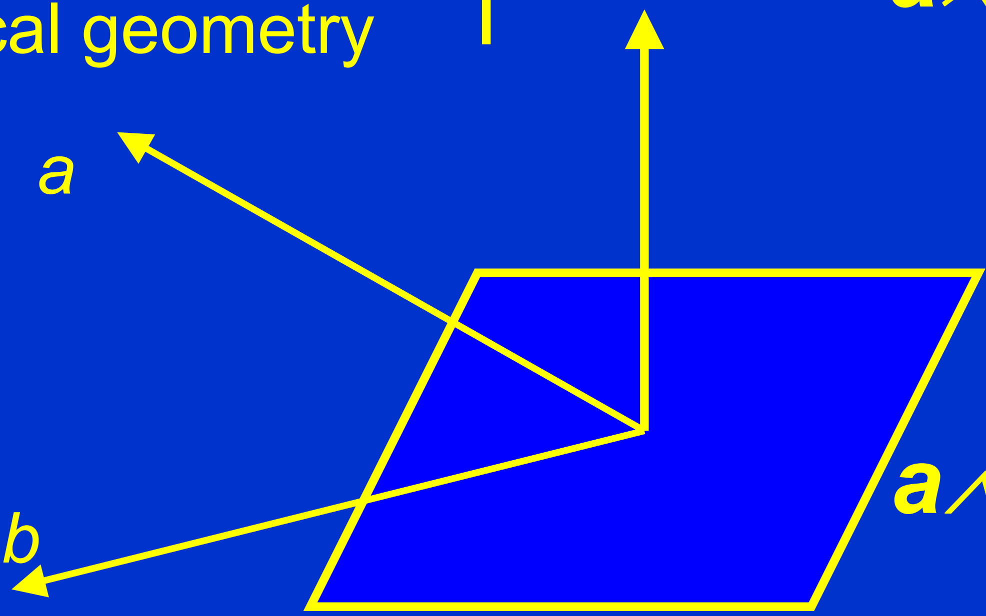
Duality

Quaternions

Vector algebra

Spherical geometry

$$\mathbf{a} \times \mathbf{b} = -i\, \mathbf{a} \wedge \mathbf{b}$$



$$\mathbf{a} \wedge \mathbf{b} = -i\, \mathbf{a} \times \mathbf{b}$$

Representations

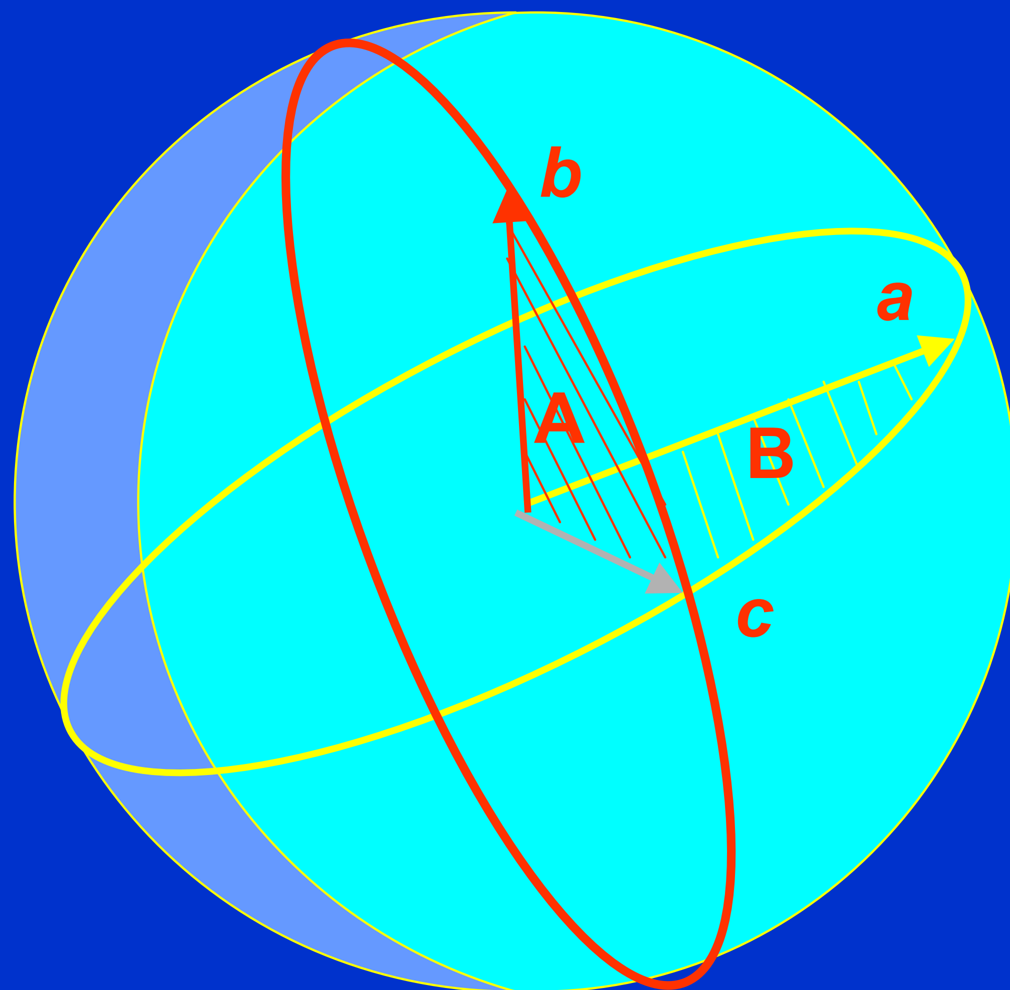
Complex analysis

Duality

Quaternions

Vector algebra

Spherical geometry



A = *bc*

B = *ca*

C = AB?

Representations

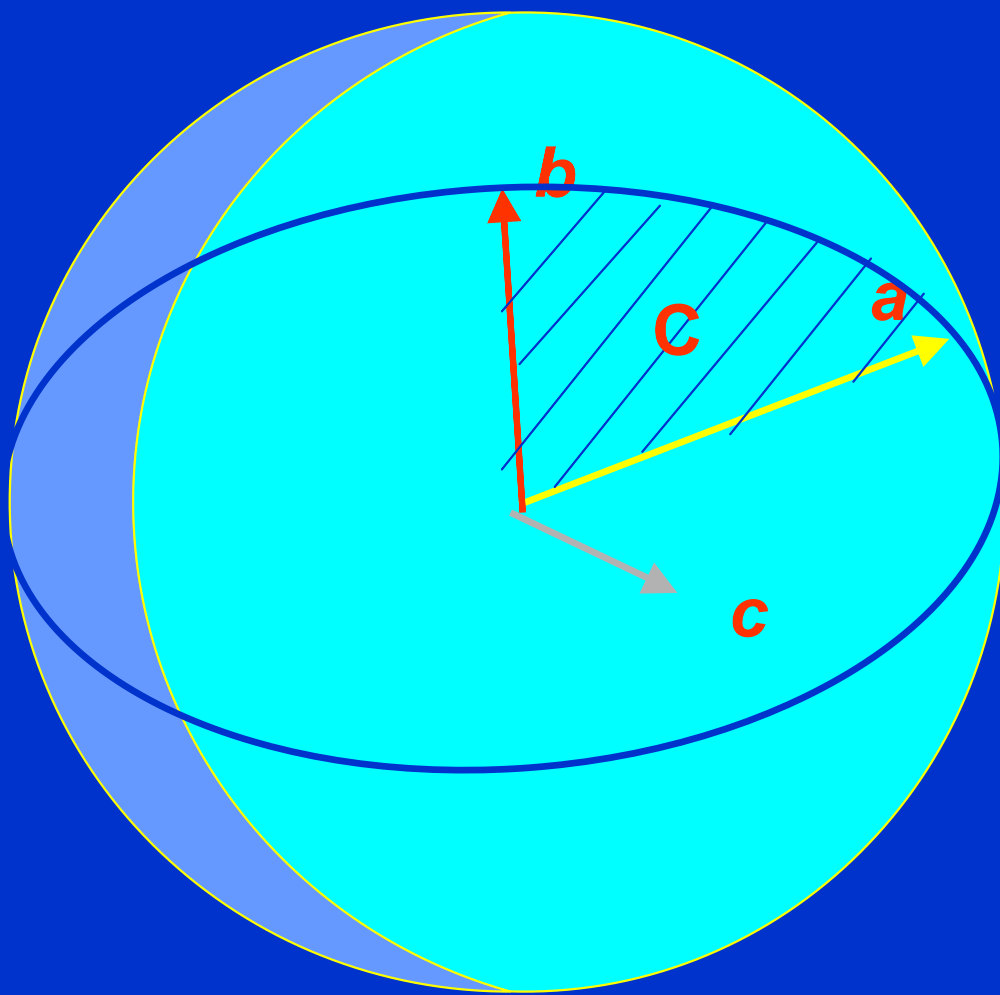
Complex analysis

Duality

Quaternions

Vector algebra

Spherical geometry



A = *bc*

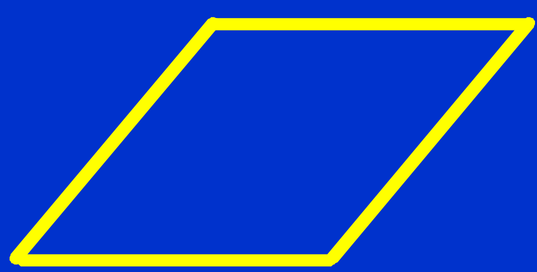
B = *ca*

C = *AB* =bcca = ba

Advantages of GA

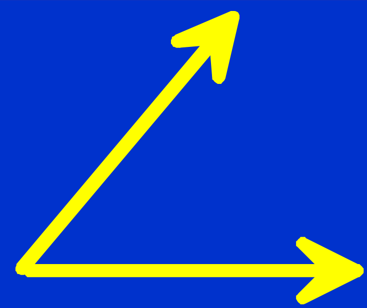
- **Unifying**
  - **compact knowledge, enhanced learning, eliminates redundancies and translation**
- **Geometrically intuitive**
- **Efficient**
  - **reduces operations, coordinate free, separation of parts**
- **Dimensionally fluid**
  - **equations across dimensions**

Bivectors

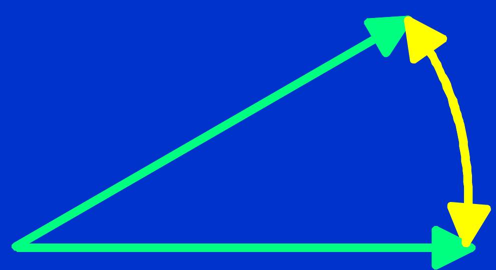


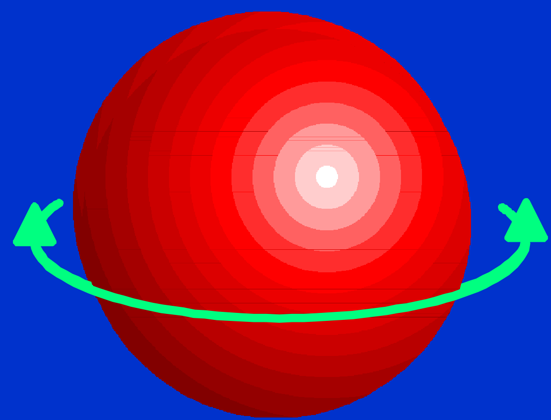
$\neq$

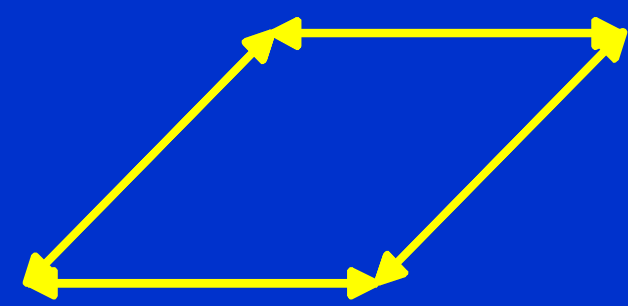
Not two vectors



Examples:







The Geometry of the Algebra

SIGGRAPH 2001, Course #53

Leo Dorst
University of Amsterdam
Amsterdam, The Netherlands
leo@science.uva.nl

Stephen Mann
University of Waterloo
Waterloo, ON, Canada
smann@cgl.uwaterloo.ca

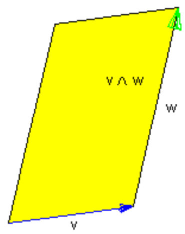
Geometric Algebra

- The geometric product ab does it all
- Algebraically, it is
  - linear
  - associative
  - non-commutative
  - invertible
- We will visualize these properties

Properties

Algebra	Geometry
anti-commutation $\frac{1}{2}(ab - ba)$	$a \wedge b$ spanning
commutation $\frac{1}{2}(ab + ba)$	$a \cdot b$ complementation perpendicularity
invertibility	orthogonalization
division	rotation

Outer product: spanning



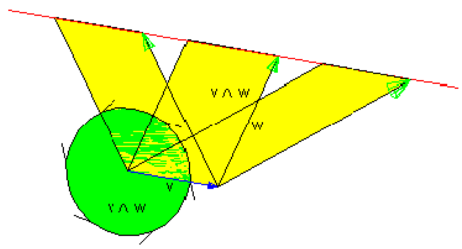
$$a \wedge b = -b \wedge a$$

- dimensionality
- attitude
- sense
- magnitude

- Equivalence

$$a \wedge (b - a) = a \wedge b$$

Amount of oriented area in a plane



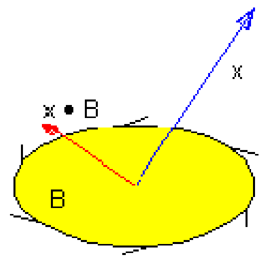
DEMOouter

- Given a, all x with same $x \wedge a$ are on a line
- Extension: $a \wedge b \wedge c$ is a volume
- $\dim(A \wedge B) = \dim(A) + \dim(B)$ (but beware of overlap)

Inner product

$$a \cdot b = b \cdot a$$

- $A \cdot B$ is part of B perpendicular to A
- Given a, all x with same $x \cdot a$ are on a hyperplane

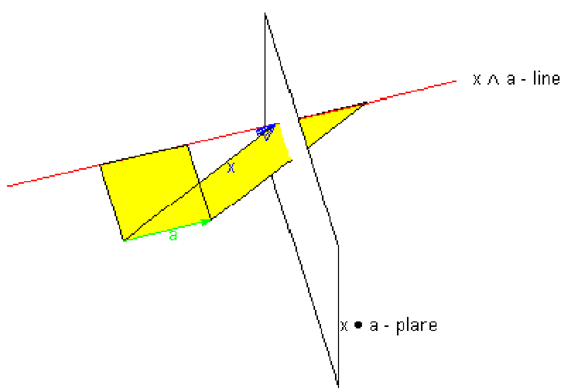


DEMOinner

- $\dim(A \cdot B) = \dim(B) — \dim(A)$

Geometric Product is Invertible

- Now put it all together:
  Given a and $x \cdot a$ and $x \wedge a$, we can reconstruct x

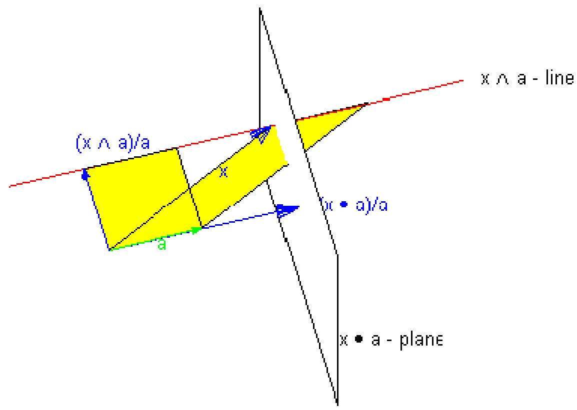


DEMOinvertible

- $xa = x \cdot a + x \wedge a$ is invertible

The Parts of the Geometric Product

- You can separate the parts: $x \cdot a$ is a scalar while $x \wedge a$ is bivector, but they don't "mix"



- What is '+' doing?

$$x = (xa)/a = (x \cdot a)/a + (x \wedge a)/a$$

Are two terms on right both vectors?

Perpendicularity

Consider $x = x_\perp + x_{||}$ relative to some vector a

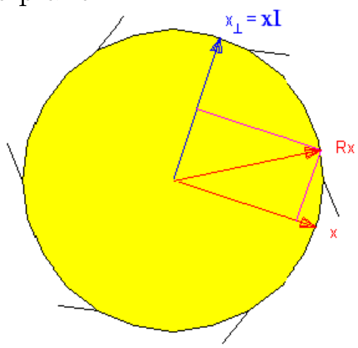
- Geometrically: $x_\perp$ is part of x perpendicular to a
- Classically: $x_\perp \wedge a = x \wedge a$ and $x_\perp \cdot a = 0$
- Geometric Algebra: $x_\perp a = x \wedge a$
  Solvable: $x_\perp = (x \wedge a)/a$

Parallel Component

- Geometrically: $x_{||}$ is part of x parallel to a
- Classically: $x_{||} \wedge a = 0$ and $x_{||} \cdot a = x \cdot a$
- Geometric Algebra: $x_{||}a = x \cdot a$
  Solvable: $x_{||} = (x \cdot a)/a$

Rotations

- Many ways to do rotations in geometric algebra
- Given x and plane I containing x (so $x \wedge I = 0$)
  Rotate x in the plane



DEMOrotdefinition

- Coordinate free view
  $$Rx = \text{bit of } x \text{ and bit of perpendicular to } x$$
  (amounts depend on rotation angle)
- Perpendicular to x in I plane (anti-clockwise) is
  $$x \cdot I = xI = -Ix$$
- Rotation as post-multiply:
  $$\mathsf{R}x = x(\cos\phi) + (xI)(\sin\phi) = x(\cos\phi + I\sin\phi)$$
- Rotation as pre-multiply:
  $$\mathsf{R}x = (\cos\phi) + (\sin\phi)(-Ix) = (\cos\phi - I\sin\phi)x$$

Complex Rotations

- Related to complex numbers

$$II = -1$$

but I has a geometrical meaning

- We can write $\boxed{\cos\phi + I\sin\phi = e^{I\phi}}$
- Each rotation plane has own bivector I
  so many "complex numbers" in space
- Bivector basis ($\mathbf{i} = e_2 \wedge e_3$, $\mathbf{j} = e_3 \wedge e_1$, $\mathbf{k} = e_1 \wedge e_2$)

$$I = \alpha\mathbf{i} + \beta\mathbf{j} + \gamma\mathbf{k}$$

Rotations in 3D

- Pick rotation plane I and (possibly non-coplanar) vector x

$$x = x_\perp + x_{||}$$

Would like to get $R_{I\phi}x = x_\perp + R_{I\phi}x_{||}$.

- $x_{||}$ rotation:
  either $e^{-I\phi}x_{||}$ or $x_{||}e^{I\phi}$ (or even $e^{-I\phi/2}x_{||}e^{I\phi/2}$)
- $x_\perp$ rotation:

$$e^{-I\phi}x_\perp = \underbrace{\cos\phi\, x_\perp}_{vector} - \underbrace{\sin\phi\,(Ix_\perp)}_{trivector}$$

$$x_\perp e^{-I\phi} = \cos\phi\, x_\perp + \sin\phi\,(x_\perp I)$$

$$\begin{aligned}
(e^{-I\phi}x_\perp)e^{I\phi} &= \cos\phi\, x_\perp e^{I\phi} - \sin\phi\, I x_\perp e^{I\phi} \\
&= \cos^2\phi\, x_\perp + \cos\phi\sin\phi\, x_\perp I \\
&\quad - \sin\phi\cos\phi\, I x_\perp - \sin^2\phi\, I x_\perp I \\
&= \cos^2\phi\, x_\perp - sin^2\phi\, I I x_\perp \\
&= (\cos^2\phi + \sin^2\phi) x_\perp \\
&= x_\perp
\end{aligned}$$

- Bottom line:

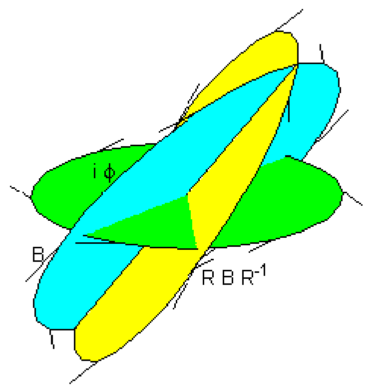
$$e^{-I\phi/2} x e^{I\phi/2} = x_{\perp} + R_{I\phi} x_{||} = R_{I\phi} x$$

Rotors

- So $R_{-I\phi} x = e^{-I\phi/2} x e^{I\phi/2}$
- Further,

$$\boxed{R_{-I\phi} X = e^{-I\phi/2} X e^{I\phi/2} = R X R^{-1}}$$

where X is any geometric object (vector, plane, volume, etc.)



DEMOrotor

- $R = e^{-I\phi/2}$ is called a *rotor*
  $R^{-1} = e^{I\phi/2}$ is called the *inverse rotor*

Quaternions

- A rotor is a (unit) quaternion
- **i**, **j**, **k** are not complex numbers, they are
  - bivectors (not vectors!)
  - rotation operators for the coordinate planes
  - basis for planes of rotation
  - an intrinsic part of the algebra

Composing Rotations

Composition of rotations through multiplication

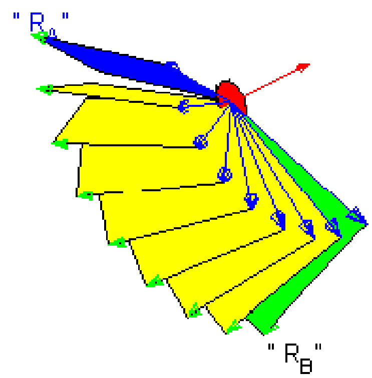
$$(\mathsf{R}_2 \circ \mathsf{R}_1)x = R_2(R_1 x R_1^{-1}) = (R_2 R_1)x(R_2 R_1)^{-1}$$

- $R_2 R_1$ is again a rotor.
  It represents the rotation $\mathsf{R}_2 \circ \mathsf{R}_1$
- Note: use geometric product to multiply rotors/quaternions
  No new product is needed

Interpolation

$$R = e^{I\phi/2} = (\underbrace{e^{I\phi/2/n} e^{I\phi/2/n} \ldots}_{n})$$

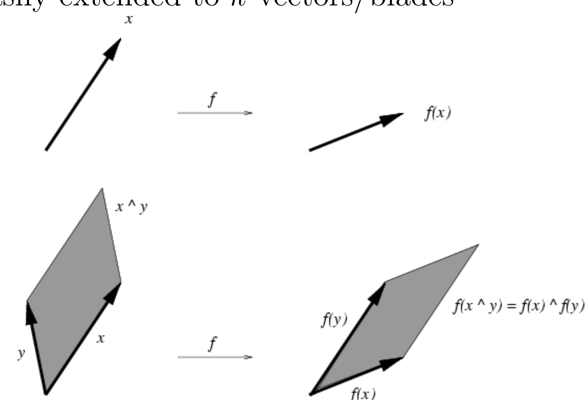
Do 1 rotation in n similar steps



DEMOinterpolation

Briefly: Linear Algebra

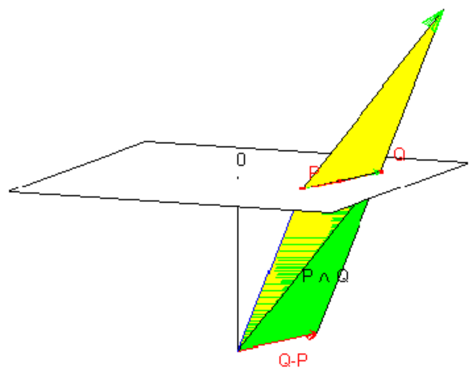
- If a linear transformation of vectors is known
  it is easily extended to k-vectors/blades



- This is called an *outermorphism*
- So linear algebra applies to higher dimensional objects

Homogeneous model

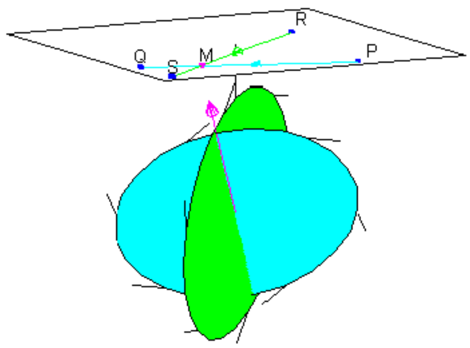
- Get affine/homogeneous spaces by using one dimension for "point at zero"
  - **Point:** $P = e + p$ such that $e \cdot p = 0$
  - **Vector:** v such that $e \cdot v = 0$
  - **Tangent plane:** bivector B such that $e \cdot B = 0$ (not a normal!)
  - **Line:** point P, point Q: $P \wedge Q = (e + p) \wedge (q - p)$
  - **Line:** direction v, point P: $P \wedge v = (e + p) \wedge v$



DEMOhomogeneous

Meet and Join

- Homogeneous line intersection requires blade intersection: *meet(A,B)*
- Dual operation, *join(A,B)*, spans lowest grade subspace of A and B.



DEMOhomogeneousmeet

For a free copy of GABLE and a geometric algebra tutorial, see
`http://www.science.uva.nl/~leo/clifford/gable.html`
`http://www.cgl.uwaterloo.ca/~smann/GABLE/`

Geometric algebra: a computational framework for geometrical applications

Leo Dorst and Stephen Mann

DRAFT April 24, 2001

Abstract

Geometric algebra is a consistent computational framework in which to define geometric primitives and their relationships. This algebraic approach contains all geometric operators and permits specification of constructions in a totally coordinate-free manner. Since it contains primitives of any dimensionality (rather than just vectors) it has no special cases: all intersections of primitives are computed with one general incidence operator. We show that the quaternion representation of rotations is also naturally contained within the framework. Models of Euclidean geometry can be made which directly represent the algebra of spheres.

1 Beyond vectors

In the usual way of defining geometrical objects in fields like computer graphics, robotics and computer vision, one uses vectors to characterize the construction. To do this effectively, the basic concept of a vector as an element of a linear space is extended by an inner product and a cross product, and some rather extraneous constructions such as homogeneous coordinates and Grassmann spaces (see [7]) to encode compactly the intersection of, for instance, offset planes in space. Many of these techniques work rather well in 3-dimensional space, although some problems have been pointed out: the difference between vectors and points, and the affine non-covariance of the normal vector as a characterization of a tangent line or tangent plane (i.e. the normal vector of a transformed plane is not the transform of the normal vector). These problems are then traditionally fixed by the introduction of certain data structures with certain combination rules; object-oriented programming can be used to implement this patch tidily.

Yet there are deeper issues in geometric programming which are still accepted as 'the way things are'. For instance, when you need to intersect linear subspaces, the intersection algorithms are split out in treatment of the various cases: lines and planes, planes and planes, lines and lines, et cetera, need to be treated in separate pieces of code. The linear algebra of the systems of equations with its vanishing determinants indicates changes in essential degeneracies, and finite and infinite intersections can be nicely unified by using homogeneous coordinates. But there seems no getting away from the necessity of separating the cases. After all, the outcomes themselves can be points, lines or planes, and those are essentially different in their further processing.

Yet this need not be so. If we could see subspaces as basic elements of computation, and do direct algebra with them, then algorithms and their implementation would not need to split their cases on dimensionality. For instance, $A \wedge B$ could be 'the subspace spanned by the spaces A and B', the expression $A \cdot B$ could be 'the part of B perpendicular to A'; and then we would always have the computation rule $(A \wedge B) \cdot C = A \cdot (B \cdot C)$ since computing the part of C perpendicular to the span of A and B can be computed in two steps, perpendicularity to B followed by perpendicularity to A. Subspaces therefore have computational rules of their own which can be used immediately, independent of how many vectors were used to span then (i.e. independent of their dimensionality). In this view, the split in cases for the intersection could be avoided, since intersection of subspaces always leads to subspaces. We should consider using this structure, since it would enormously simplify the specification of geometric programs.

This paper intends to convince you that subspaces form an algebra with well-defined products which have direct geometric significance. That algebra can then be used as a language for geometry, and we claim that it is a better choice than a language always reducing everything to vectors (which are just 1-dimensional subspaces). It comes as a bit of a surprise that there is really one basic product between subspaces that forms the basis for such an algebra, namely the *geometric product*. The algebra is then what mathematicians call a Clifford algebra. But for applications, it is often very convenient to consider 'components' of this geometric product; this gives us sensible extensions, to subspaces, of the inner product (computing measures of perpendicularity), the cross product (computing measures of parallelness), and the meet and join (computing intersection and union of subspaces). When used in such an obviously geometrical way, the term *geometric algebra* is preferred to describe the field.

In this paper, we will use the basic products of geometric algebra to describe all familiar elementary constructions of basic geometric objects and their quantitative relationships. The goal is to show you that this can be done, and that it is compact, directly computational, and transcends the dimensionality of subspaces. We will not use geometric algebra to develop new algorithms for graphics; but we hope you to convince you that some of the lower level algorithmic aspects can be taken care of in an automatic way, without exceptions or hidden degenerate cases by using geometric algebra as a language – instead of only its vector algebra part as in the usual approach.

2 Subspaces as elements of computation

As in the classical approach, we start with a real vector space R^n which we use to denote 1-dimensional directed magnitudes. Typical usage would be to employ a vector to denote a translation in such a space, to establish the location of a point of interest. (Points are not vectors, but their locations are.) Another usage is to denote the velocity of a moving point. (Points are not vectors, but their velocities are.) We now want to extend this capability of indicating directed magnitudes to higher-dimensional directions such as facets of objects, or tangent planes. In doing so, we will find that we have automatically encoded the algebraic properties of multi-point objects such as line segments or circles. This is rather surprising, and not at all obvious from the start. For educational reasons, we will start with the simplest subspaces: the 'proper' subspaces of a linear vector space which are lines, planes, etcetera through the origin, and develop their algebra of spanning and perpendicularity measures. Only in Section refmodels do we show some of the considerable power of the products when used in the context of models of geometries.

2.1 Vectors

So we start with a real m-dimensional linear space V^m, of which the elements are called *vectors*. They can be added, with real coefficients, in the usual way to produce new vectors.

We will always view vectors geometrically: a vector will denote a '1-dimensional direction element', with a certain 'attitude' or 'stance' in space, and a 'magnitude', a measure of length in that direction. These properties are well characterized by calling a vector a 'directed line element', as long as we mentally associate an orientation and magnitude with it: $\mathbf{v}$ is not the same as $-\mathbf{v}$ or $2\mathbf{v}$.

2.2 The outer product

In geometric algebra, higher-dimensional oriented subspaces are also basic elements of computation. They are called *blades*, and we use the term k-*blade* for a k-dimensional homogeneous subspace. So a vector is a 1-blade. (Again, we first focus on 'proper' linear subspaces, i.e. subspaces which contain the origin: the 1-dimensional homogeneous subspaces are lines through the origin, the 2-dimensional homogeneous subspaces are planes through the origin, etc.)

A common way of constructing a blade is from vectors, using a product that constructs the span of vectors. This product is called the *outer product* (sometimes the *wedge product*) and denoted by $\wedge$. It is codified by its algebraic properties, which have been chosen to make sure we indeed get m-dimensional space elements with an appropriate magnitude (area element for $m = 2$, volume elements for $m = 3$). As you have seen in linear algebra, such magnitudes are determinants of matrices representing the basis of vectors spanning them. But such a definition would be too specifically dependent on that matrix representation. Mathematically, a determinant is viewed as an anti-symmetric linear scalar-valued function of its vector arguments. That gives the clue to the rather abstract definition of the outer product in geometric algebra:

> The outer product of vectors $\mathbf{a}_1, \cdots, \mathbf{a}_k$ is anti-symmetric, associative and linear in its arguments. It is denoted as $\mathbf{a}_1 \wedge \cdots \wedge \mathbf{a}_k$, and called a k-blade.

The only thing that is different from a determinant is that the outer product is *not* forced to be scalar-valued; and this gives it the capability of representing the 'attitude' of a k-dimensional subspace element as well as its magnitude.

2.3 2-blades in 3-dimensional space

Let us see how this works in the geometric algebra of a 3-dimensional space V^3. For convenience, let us choose a basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ in this space, relative to which we denote any vector (there is no need to choose this basis orthonormally – we have not mentioned the inner product yet – but you can think of it as such if you like). Now let us compute $\mathbf{a} \wedge \mathbf{b}$ for $\mathbf{a} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3$ and $\mathbf{b} = b_1\mathbf{e}_1 + b_2\mathbf{e}_2 + b_3\mathbf{e}_3$. By linearity, we can write this as the sum of six terms of the form $a_1 b_2 \mathbf{e}_1 \wedge \mathbf{e}_2$ or $a_1 b_1 \mathbf{e}_1 \wedge \mathbf{e}_1$. By anti-symmetry, the outer product of any vector with itself must be zero, so the term with $a_1 b_1 \mathbf{e}_1 \wedge \mathbf{e}_1$ and other similar terms disappear. Also by anti-symmetry, $\mathbf{e}_2 \wedge \mathbf{e}_1 = -\mathbf{e}_1 \wedge \mathbf{e}_2$, so some terms can be grouped. You may verify that the final result is:

$$
\begin{aligned}
&\mathbf{a} \wedge \mathbf{b} = \\
&\quad = (a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3) \wedge (b_1\mathbf{e}_1 + b_2\mathbf{e}_2 + b_3\mathbf{e}_3) \\
&\quad = (a_1 b_2 - a_2 b_1)\, \mathbf{e}_1 \wedge \mathbf{e}_2 + (a_2 b_3 - a_3 b_2)\, \mathbf{e}_2 \wedge \mathbf{e}_3 + (a_3 b_1 - a_1 b_3)\, \mathbf{e}_3 \wedge \mathbf{e}_1 \qquad (1)
\end{aligned}
$$

We cannot simplify this further. Apparently, the axioms of the outer product permit us to decompose any 2-blade in 3-dimensional space onto a basis of 3 elements. This '2-blade basis' (also called 'bivector basis) $\{\mathbf{e}_1 \wedge \mathbf{e}_2, \mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1\}$ consists

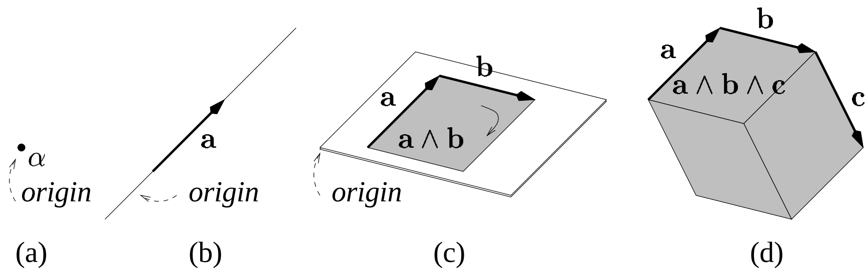


Figure 1: *Spanning proper subspaces using the outer product.*

of 2-blades spanned by the basis vectors. Linearity of the outer product implies that the set of 2-blades forms a linear space on this basis. We will interpret this as the space of all *plane elements* or *area elements*. Let us show that they have indeed the correct magnitude for an area element. That is particularly clear if we choose a particular orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, chosen such that $\mathbf{a}$ lies in the $\mathbf{e}_1$-direction, and $\mathbf{b}$ lies in the $(\mathbf{e}_1, \mathbf{e}_2)$-plane. Then $\mathbf{a} = a\mathbf{e}_1$, $\mathbf{b} = b\cos\phi\,\mathbf{e}_1 + b\sin\phi\,\mathbf{e}_2$ (with ϕ the angle from $\mathbf{a}$ to $\mathbf{b}$), so that

$$\mathbf{a} \wedge \mathbf{b} = (a\,b\,\sin\phi)\,\mathbf{e}_1 \wedge \mathbf{e}_2 \tag{2}$$

This single result contains both the correct magnitude of the area $a\,b\,\sin\phi$ spanned by $\mathbf{a}$ and $\mathbf{b}$, and the plane in which it resides – for we should learn to read $\mathbf{e}_1 \wedge \mathbf{e}_2$ as 'the unit directed area element of the $(\mathbf{e}_1, \mathbf{e}_2)$-plane'. Since we can always adapt our coordinates to vectors in this way, this result is universally valid: $\mathbf{a} \wedge \mathbf{b}$ is an area element of the plane spanned by $\mathbf{a}$ and $\mathbf{b}$.

You can visualize this as the parallelogram spanned by $\mathbf{a}$ and $\mathbf{b}$, but you should be a bit careful: the *shape* of the area element is not defined in $\mathbf{a} \wedge \mathbf{b}$. For instance, by the properties of the outer product, $\mathbf{a} \wedge \mathbf{b} = \mathbf{a} \wedge (\mathbf{b} + \lambda\mathbf{a})$, for any λ, so the parallelogram can be sheared. Also, the area element is free to translate: the sum of the area elements $\frac{1}{4}(\mathbf{a} \wedge \mathbf{b})$, $\frac{1}{4}(\mathbf{b} \wedge (-\mathbf{a}))$, $\frac{1}{4}((-\mathbf{a}) \wedge (-\mathbf{b}))$, $\frac{1}{4}((-\mathbf{b}) \wedge \mathbf{a})$ equals $\mathbf{a} \wedge \mathbf{b}$; drawing this equation shows that we should imagine the area element to have no specific location in its plane. You may also verify that an orthogonal transformation of $\mathbf{a}$ and $\mathbf{b}$ in their common plane (such as a rotation in that plane) leaves $\mathbf{a} \wedge \mathbf{b}$ unchanged. (This is obvious once you know the result for determinants and note that $\mathbf{a} \wedge \mathbf{b}$ can always be expressed as in eq.(1), but we will revisit its deeper meaning in Section 7).

It is important to realize that the 2-blades have an existence of their own, independent of any vectors that one might use to define them; that is reflected in the fact that they are not parallelograms. Planes (or, more precisely, plane elements) are nouns in our computational geometrical language, of the same basic nature as vectors (or line elements).

2.4 Volumes as 3-blades

We can also form the outer product of *three* vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$. Considering each of those decomposed onto their 3 components on some basis in our 3-dimensional space (as above), we obtain terms of three different types, depending on how many common components occur: terms like $a_1 b_1 c_1\, \mathbf{e}_1 \wedge \mathbf{e}_1 \wedge \mathbf{e}_1$, like $a_1 b_1 c_2\, \mathbf{e}_1 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2$, and like $a_1 b_2 c_3\, \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$. Because of associativity and anti-symmetry, only the last type survives, in all its permutations. The final result is:

$$\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} = (a_1 b_2 c_3 - a_1 b_3 c_2 + a_2 b_1 c_3 - a_2 b_3 c_1 + a_3 b_1 c_2 - a_3 b_2 c_1)\, \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3.$$

The scalar factor is the determinant of the matrix with columns $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, which is proportional to the signed volume spanned by them (as is well known from linear algebra). The term $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ is the denotation of which volume is used as unit: that spanned by $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$. The order of the vectors gives its orientation, so this is a 'signed volume'. In 3-dimensional space, there is not really any other choice for the construction of volumes than (possibly negative) multiples of this volume. But in higher dimensional spaces, the attitude of the volume element needs to be indicated just as much as we needed to denote the attitude of planes in 3-space.

2.5 Linear dependence

Note that if the three vectors are linearly dependent, they satisfy:

$$\mathbf{a},\mathbf{b},\mathbf{c} \text{ linearly dependent} \quad \iff \quad \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} = 0.$$

We interpret the latter immediately as the geometric statement that the vectors span a zero volume. This makes linear dependence a computational property rather than a predicate: three vectors can be 'almost linearly dependent'. The magnitude of $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$ obviously involves the determinant of the matrix $(\mathbf{a}\ \mathbf{b}\ \mathbf{c})$, so this view corresponds with the usual computation of determinants to check degeneracy.

2.6 The pseudoscalar as hypervolume

Forming the outer product of *four* vectors $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} \wedge \mathbf{d}$ in 3-dimensional space will always produce zero (since they must be linearly dependent). To see this, just decompose the vectors on some basis (for instance, the fourth vector on a basis formed by the other 3), and apply the outer product. Since $(\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c})$ is proportional to $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$, multiplication by $\mathbf{d}$ will always lead to terms like $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_1$, in which at least two vectors are the same. Associativity and anti-symmetry then makes all terms equal to zero.

The highest order blade which is non-zero in an m-dimensional space is therefore an m-blade. Such a blade, representing an m-dimensional volume element, is called a *pseudoscalar* for that space (for historical reasons); unfortunately a rather abstract term for the elementary geometric concept of 'hypervolume element'.

The dimensionality of a k-blade is the number of vector factors that span it; this is usually called the *grade* of the blade. It obeys the simple rule:

$$\mathsf{grade}\,(\mathbf{A} \wedge \mathbf{B}) = \mathsf{grade}\,(\mathbf{A}) + \mathsf{grade}\,(\mathbf{B})\,. \tag{3}$$

Of course the outcome may be 0, so this zero element of the algebra should be seen as an element of arbitrary grade. There is then no need to distinguish separate zero scalars, zero vectors, zero 2-blades.

2.7 Scalars as subspaces

To make scalars fully admissible elements of the algebra we have so far, we can define the outer product of two scalars, and a scalar and a vector, through identifying it with the familiar scalar product in the vector space we started with:

$$\alpha \wedge \beta = \alpha\,\beta \quad \text{and} \quad \alpha \wedge \mathbf{v} = \alpha\,\mathbf{v}$$

This automatically extends (by associativity) to the outer product of scalars with higher order blades.

We will denote scalars mostly by Greek lower case letters. Since they are constructed by the outer product of zero vectors, we can interpret the scalars as the representation in geometric algebra of 0-dimensional subspace elements, i.e. as a *weighted points at the origin* – or maybe you prefer 'charged', since the weight can be negative. This is indeed consistent, we will get back to that when intersecting subspaces in Section 4.

2.8 The Grassmann algebra of 3-space

Collating what we have so far, we have constructed a geometrically significant algebra containing only two operations: the addition $+$ and the outer multiplication $\wedge$ (subsuming the usual scalar multiplication). Starting from scalars and a 3-dimensional vector space we have generated a 3-dimensional space of 2-blades, and a 1-dimensional space of 3-blades (since all volumes are proportional to each other). In total, therefore, we have a set of elements which naturally group by their dimensionality. Choosing some basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, we can write what we have as spanned by the set:

$$\left\{ \underbrace{1}_{\textit{scalars}}, \underbrace{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3}_{\textit{vector space}}, \underbrace{\mathbf{e}_1 \wedge \mathbf{e}_2,\ \mathbf{e}_2 \wedge \mathbf{e}_3,\ \mathbf{e}_3 \wedge \mathbf{e}_1}_{\textit{bivector space}}, \underbrace{\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3}_{\textit{trivector space}} \right\} \quad (4)$$

Every k-blade formed by $\wedge$ can be decomposed on the k-vector basis using $+$. The 'dimensionality' k is often called the *grade* or *step* of the k-blade or k-vector, reserving the term *dimension* for that of the vector space which generated them. A k-blade represents a k-dimensional oriented subspace element.

If we allow the scalar-weighted addition of arbitrary elements in this set of basis blades, we get an 8-dimensional linear space from the original 3-dimensional vector space. This space, with $+$ and $\wedge$ as operations, is called the *Grassmann algebra* of 3-space.

We have no interpretation (yet) for mixed-grade terms such as $1+\mathbf{e}_1$. Actually, even addition of elements of the same grade is hard to interpret in spaces of more than 3 dimensions, since it easily leads to elements that cannot be decomposed using the outer product – so to non-blades, i.e. objects that cannot be 'spanned' by vectors. (For instance, $\mathbf{e}_1 \wedge \mathbf{e}_2 + \mathbf{e}_3 \wedge \mathbf{e}_4$ in 4-space cannot be written in the form $\mathbf{a} \wedge \mathbf{b}$ – try it!) The general term for the sum of k-blades (for the same k) is k-*vector*, and the general term for the mixed-grade elements permitted in Grassmann algebra is *multivector*.

2.9 Many blades

¿From the way it is constructed through the anti-symmetric product, it should be clear that the k-dimensional subspaces of an m-dimensional space have a basis which consists of a number of independent elements equal to the number of ways one can take k distinct indices from a set of m indices. That is

> The linear space of k-vectors in m-space is $\binom{m}{k}$-dimensional.

Adding them all up, we find:

> The linear space of all subspaces of an m-dimensional vector space is 2^m-dimensional.

To have a basis for all possible subspaces (through the origin) in 3-dimensional space takes $2^3 = 8$ elements, such as in eq.(4). You can characterize an element X of that space therefore by a 8×1 matrix $[X]$. Since the outer product by another element vector A is linear, $A \wedge X$ can be written as the action of a linear operator

$A^\wedge$ on X, and hence be represented as a matrix multiplication $[A^\wedge]\,[X]$, with $[A^\wedge]$ an 8×8 matrix. This is not a particularly efficient representation, but it shows that this algebra of $+$ and $\wedge$ on a vector space is just a special linear algebra; a fact which may give you some confidence that it is at least consistent.

When they just learn about this algebra, most people are put off by how many blades there are, and some have rejected the practical use of geometric algebra because of its exponentially large basis. This is a legitimate concern, and the implementation just sketched obviously does not scale well with dimensionality. For now, a helpful view may be to see this 2^m-dimensional basis as a cabinet in which all relationships which we may care to compute in the course of our computations in m-dimensional space can be filed properly: k-point relationships in the $\binom{m}{k}$ files in the k-th drawer. And the files themselves have clear computational relationships (we have seen the outer product, more will follow). This should be compared to the usual way in which such k-point relationships are made whenever they are needed, but not preserved in a structural way relating them algebraically to the other relationships of the application. This simile suggests that there might be some potential gain in building up the overall structure rather than reinventing it several times along the way, as long as we make sure that this organization does not affect the efficiency of individual computations too much. This paper should provide you with sufficient material to ponder this new possibility.

3 Relative subspaces measures

The outer product gives computational meaning to the notion of 'spanning subspaces'. It does not use any metric structure which we may have available for our original vector space V^m. The familiar inner product of vectors in a vector space *does* use the metric – in fact, it *defines* the metric, since it gives a bilinear form returning a scalar value $\mathbf{a} \cdot \mathbf{b}$ for each pair of vectors, which can be used to defined the distance measure $\sqrt{(\mathbf{a} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b})}$. Now that vectors are viewed as representatives of 1-dimensional subspaces, we of course want to extend this metric capability to arbitrary subspaces. This leads to the *scalar product*, and its meshing with the outer product gives a generalized *inner product* between blades.

3.1 The scalar product: a metric for blades

Between two blades $\mathbf{A}_k$ and $\mathbf{B}_k$ of the same grade k, we can define a metric measure. The most computational way of doing so is to span each of the blades by k vectors: $\mathbf{A}_k = \mathbf{a}_1 \wedge \mathbf{a}_q \wedge \cdots \wedge \mathbf{a}_k$ and $\mathbf{B}_k = \mathbf{b}_1 \wedge \mathbf{b}_q \wedge \cdots \wedge \mathbf{b}_k$. Then the scalar product between them is defined as:

$$\mathbf{A}_k * \mathbf{B}_k \equiv \begin{vmatrix} \mathbf{a}_1 \cdot \mathbf{b}_k & \mathbf{a}_1 \cdot \mathbf{b}_{k-1} & \cdots & \mathbf{a}_1 \cdot \mathbf{b}_1 \\ \mathbf{a}_2 \cdot \mathbf{b}_k & \mathbf{a}_2 \cdot \mathbf{b}_{k-1} & \cdots & \mathbf{a}_2 \cdot \mathbf{b}_1 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_k \cdot \mathbf{b}_k & \mathbf{a}_k \cdot \mathbf{b}_{k-1} & \cdots & \mathbf{a}_k \cdot \mathbf{b}_1 \end{vmatrix} \tag{5}$$

The unfortunate order of the factors was chosen historically. We get a nicer form if we introduce an operation that reverses a factorization, for instance $\mathbf{A} = \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \mathbf{a}_3$ would become $\mathbf{a}_3 \wedge \mathbf{a}_2 \wedge \mathbf{a}_1$. (We need this for other purposes as well, or we would have preferred to fix the scalar product.) Due to the anti-symmetry of the outer product, these differ only by a sign factor, for a k-blade a sign of $(-1)^{\frac{1}{2}k(k-1)}$. We denote it by a tilde, so: $\widetilde{\mathbf{A}} = \mathbf{a}_3 \wedge \mathbf{a}_2 \wedge \mathbf{a}_1 = -\mathbf{A}$. Now $\widetilde{A} * \mathbf{B}$ has nicely matching coefficients.

The value of $\widetilde{\mathbf{A}} * \mathbf{B}$ is independent of the factorization of $\mathbf{A}$ and $\mathbf{B}$, as you may verify by the properties of determinants: adding a multiple of, say $\mathbf{a}_2$ to $\mathbf{a}_1$ leaves the blade $\mathbf{A}$ unchanged, so it should give the same answer. In $\widetilde{\mathbf{A}} \cdot \mathbf{B}$, it leads to addition of a multiple of the second column to the first, and this indeed leaves the determinant unchanged – the two anti-symmetries in the definitions of $\wedge$ and $*$ match well. The value of $\widetilde{\mathbf{A}} * \mathbf{B}$ is proportional to the cosine of the angle of the two subspaces – if a rotation exists that rotates one into the other, otherwise it is zero. The definition is extended to blades of different grade by setting $\mathbf{A} * \mathbf{B} = 0$ whenever the grades are different. So no scalar metric comparison is possible between such different subspaces (but for them we have the inner product of the next section).

The scalar product of a subspace with itself gives us the *norm* of the subspace, defined as [1]:

$$|\mathbf{A}| = \sqrt{\widetilde{\mathbf{A}} * \mathbf{A}} \tag{6}$$

For a 2-blade $\mathbf{A} = \mathbf{a}_1 \wedge \mathbf{a}_2$, with an angle of ϕ between $\mathbf{a}_1$ and $\mathbf{a}_2$, you may verify that this gives $|\mathbf{A}| = |\mathbf{a}_1|\,|\mathbf{a}_2|\,|\sin\phi|$, the absolute value of the area measure, precisely what one would hope.

3.2 The inner product

The geometric nature of blades means that there are relationships between the metric measures of different grades: for instance, the angle two 2-blades make is related to that of two properly chosen vectors in their planes (see Figure 2). We

[1] This works only in a Euclidean metric in a real vector space; in other metrics one should define the 'norm squared' and avoid the square root.

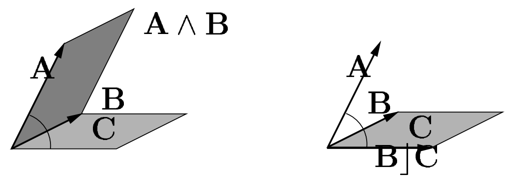


Figure 2: *The metric relationship between different spans.*

should therefore be capable of relating those numerically. If a blade is spanned as $\mathbf{A} \wedge \mathbf{B}$, and we are interested in its measure relative to $\mathbf{C}$ we compute $(\mathbf{A} \wedge \mathbf{B}) * \mathbf{C}$; but we should be able to find a similar measure between the subblade $\mathbf{A}$, and some subblade of $\mathbf{C}$, which is '$\mathbf{C}$ with $\mathbf{B}$ taken out'. This can be used to define a new product, through:

$$(\mathbf{A} \wedge \mathbf{B}) * \mathbf{C} = \mathbf{A} * (\mathbf{B} \cdot \mathbf{C}), \quad \text{for all } \mathbf{C} \tag{7}$$

The blade $\mathbf{B} \cdot \mathbf{C}$ is the *inner product* of $\mathbf{B}$ and $\mathbf{C}$. Its grade is the difference of the grades of $\mathbf{C}$ and $\mathbf{B}$ (since it should equal the grade of $\mathbf{A}$ in the definition). The inner product can be interpreted more directly as

> $\mathbf{B} \cdot \mathbf{C}$ is the blade representing the largest subspace which is contained in the subspace $\mathbf{C}$ and which is perpendicular to the subspace $\mathbf{B}$; it is linear in $\mathbf{B}$ and $\mathbf{C}$; it coincides with the usual inner product $\mathbf{b} \cdot \mathbf{c}$ of V^m when computed for vectors $\mathbf{b}$ and $\mathbf{c}$.

The above determines the inner product uniquely[2]. It turns out not to be symmetrical (as one would expect since the definition is asymmetrical) and also not associative. But we do demand linearity, to make it computable between any two elements in our linear space (not just blades).

For later use, we just give the rules by which to compute the resulting inner product for arbitrary blades, omitting their derivation. Then we will do some examples to convince you that it does what we want it to do. In the following α, β are scalars, $\mathbf{a}$ and $\mathbf{b}$ vectors and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ blades of arbitrary order. We give the rules in a slightly redundant form, for convenience in evaluating expressions.

$$\text{scalars} \qquad \alpha \cdot \beta = \alpha \wedge \beta \tag{8}$$

[2]The resulting inner product differs slightly from the inner product commonly used in the geometric algebra literature. Our inner product has a cleaner geometric semantics, and more compact mathematical properties, and that makes it better suited to computer science. It is sometimes called the *contraction*, and denoted as $\mathbf{B} \rfloor \mathbf{C}$ rather than $\mathbf{B} \cdot \mathbf{C}$. The two inner products can be expressed in terms of each other, so this is not a severely divisive issue. They 'algebraify' the same geometric concepts, in just slightly different ways.

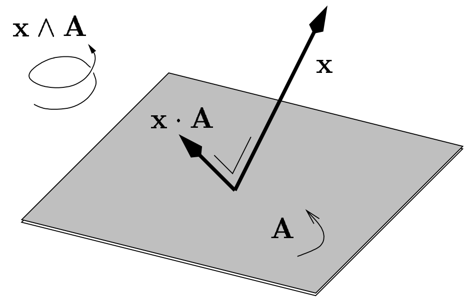


Figure 3: *The definition of the inner product of blades XXX where referred?.*

vector and scalar	$\mathbf{a} \cdot \beta = 0$	(9)
scalar and vector	$\alpha \cdot \mathbf{b} = \alpha \wedge \mathbf{b}$	(10)
vectors	$\mathbf{a} \cdot \mathbf{b}$ is the usual inner product in V^m	(11)
vector and blade	$\mathbf{a} \cdot (\mathbf{b} \wedge \mathbf{B}) = (\mathbf{a} \cdot \mathbf{b}) \wedge \mathbf{B} - \mathbf{b} \wedge (\mathbf{a} \cdot \mathbf{B})$	(12)
blades	$(\mathbf{A} \wedge \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C})$	(13)
distributivity 1	$\mathbf{A} \cdot (\mathbf{B} + \mathbf{C}) = \mathbf{A} \cdot \mathbf{B} + \mathbf{A} \cdot \mathbf{C}$	(14)
distributivity 2	$(\mathbf{A} + \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot \mathbf{C} + \mathbf{B} \cdot \mathbf{C}$	(15)

It should be emphasized that the inner product is not associative. For instance, $\mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{c}) = 0$ since the second argument is a scalar; but $(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{c} = \alpha \mathbf{c}$ (with $\alpha = \mathbf{a} \cdot \mathbf{b}$) is a vector. Neither is the inner product symmetrical, as the scalar/vector rules show.

3.3 Perpendicularity and duality

Having the inner product expands our capabilities in geometric computations. It enables manipulation of expressions involving ‘spanning’ to being about ‘perpendicularity’ and vice versa. Such ‘dual’ formulations turn out to be very convenient. We briefly develop intuition and basic conversion expressions for these manipulations.

- *perpendicularity*
  We define the concept of perpendicularity through the inner product:

  $$\mathbf{a} \text{ perpendicular to } \mathbf{A} \quad \Longleftrightarrow \quad \mathbf{a} \cdot \mathbf{A} = 0,$$

  It is then easy to prove that, for general blades $\mathbf{A}$, the construction $\mathbf{A} \cdot \mathbf{B}$ is indeed perpendicular to $\mathbf{A}$, as we suggested in the previous section. For any vector $\mathbf{a}$ satisfies $\mathbf{a} \cdot (\mathbf{A} \cdot \mathbf{B}) = (\mathbf{a} \wedge \mathbf{A}) \cdot \mathbf{B}$. But if $\mathbf{a}$ is in $\mathbf{A}$ it must be linearly dependent on the spanning vectors, so $\mathbf{a} \wedge \mathbf{A} = 0$. Therefore $\mathbf{a} \cdot (\mathbf{A} \cdot \mathbf{B}) = 0$ for any $\mathbf{a}$ in $\mathbf{A}$. So any vector in $\mathbf{A}$ is perpendicular to $\mathbf{A} \cdot \mathbf{B}$.

- *orthogonal complement and dual*
  If we take the inner product of a blade relative to the volume element of the space it resides in (i.e. relative to the pseudoscalar of the space), we get the whole subspace perpendicular to it. This is how *duality* sits in geometric algebra: it is simply taking an orthogonal complement. A good example in a 3-dimensional Euclidean space is the dual of a 2-blade (or bivector). Using an orthonormal basis $\{\mathbf{e}_i\}_{i=1}^3$ and the corresponding bivector basis, we write: $\mathbf{B} = b_1 \mathbf{e}_2 \wedge \mathbf{e}_3 + b_2 \mathbf{e}_3 \wedge \mathbf{e}_1 + b_3 \mathbf{e}_3 \wedge \mathbf{e}_2$. We take the dual relative to the space with volume element $\mathbf{I}_3 \equiv \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ (i.e. the 'right-handed volume' formed by using a right-handed basis). Any scalar multiple would do, but it turns out that the best definition is to use the *reverse* of $\mathbf{I}_3$ to define the dual (since that generalizes to higher dimensions; here $\widetilde{\mathbf{I}}_3 = -\mathbf{I}_3$). The subspace of $\mathbf{I}_3$ dual to $\mathbf{B}$ is then:

$$\begin{aligned} \mathbf{B} \cdot \widetilde{\mathbf{I}}_3 &= (b_1 \mathbf{e}_2 \wedge \mathbf{e}_3 + b_2 \mathbf{e}_3 \wedge \mathbf{e}_1 + b_3 \mathbf{e}_1 \wedge \mathbf{e}_2) \cdot (\mathbf{e}_3 \wedge \mathbf{e}_2 \wedge \mathbf{e}_1) \\ &= b_1 \mathbf{e}_1 + b_2 \mathbf{e}_2 + b_3 \mathbf{e}_3. \end{aligned} \tag{16}$$

  This is a vector, and we recognize it (in this Euclidean space) as the *normal vector* to the planar subspace represented by $\mathbf{B}$. So we have normal vectors in geometric algebra as the duals of 2-blades, if we would want them (but we will see in Section 7.3 why we prefer the direct representation of a planar subspace by a 2-blade rather than the indirect representation by normal vectors).

  If it is clear from context relative to which pseudoscalar $\mathbf{I}$ the dual is taken, we will use the convenient shorthand $\mathbf{B}^*$ for $\mathbf{B} \cdot \widetilde{\mathbf{I}}$.

- *duality relationships*
  Going over to a dual representation involves translating formulas given in terms of spanning to formulas using perpendicularity. An example is the specification of a plane in 3-space given its 2-blade $\mathbf{B}$. On the one hand, all vectors in the plane satisfy $\mathbf{x} \wedge \mathbf{B} = 0$ (zero volume spanned with the 2-blade); but dually they satisfy $\mathbf{x} \cdot \mathbf{B}^* = 0$ (perpendicular to the normal vector). This is an example of a more general duality relationship between blades, which we state without proof. Let $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{I}$ be blades, with $\mathbf{A}$ contained in $\mathbf{I}$ (this is essential). Then:

$$(\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{I} = \mathbf{A} \wedge (\mathbf{B} \cdot \mathbf{I}) \quad \text{if } \mathbf{A} \subseteq \mathbf{I}. \tag{17}$$

Remember also the universally valid eq.(13)

$$(\mathbf{A} \wedge \mathbf{B}) \cdot \mathbf{I} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{I}). \tag{18}$$

Together, these equations allow the change to a 'dual perspective' converting spanning to orthogonality and *vice versa*, permitting more flexible interpretation of equations.

Let us use these to verify the motivating example above in full detail. In a 3-dimensional space with pseudoscalar $\mathbf{I}_3$, the equation $\mathbf{x} \wedge \mathbf{B} = 0$ (meaning that $\mathbf{x}$ is in the 2-dimensional subspace determined by $\mathbf{B}$) can be dualized to $0 = (\mathbf{x} \wedge \mathbf{B}) \cdot \widetilde{\mathbf{I}}_3 = \mathbf{x} \cdot (\mathbf{B} \cdot \widetilde{\mathbf{I}}_3)$. This characterizes the vectors in the $\mathbf{B}$-plane through its normal vector $\mathbf{n} \equiv \mathbf{B} \cdot \widetilde{\mathbf{I}}_3 = \mathbf{B}^*$. It is the familiar 'normal equation' of the plane, and identical to the common way to represent a plane by its normal vector $\mathbf{n}$.

In general, we will say that a blade $\mathbf{B}$ represents a subspace $\mathcal{B}$ of vectors $\mathbf{x}$ if

$$\mathbf{x} \in \mathcal{B} \iff \mathbf{x} \wedge \mathbf{B} = 0 \tag{19}$$

and that a blade $\mathbf{B}^*$ *dually represents* the subspace $\mathcal{B}$ if

$$\mathbf{x} \in \mathcal{B} \iff \mathbf{x} \cdot \mathbf{B}^* = 0. \tag{20}$$

Switching between the two standpoints is done by the duality relations above.

- *the cross product*
  Classical computations with vectors in 3-space often use the cross product, which produces from two vectors $\mathbf{a}$ and $\mathbf{b}$ a new vector $\mathbf{a} \times \mathbf{b}$ perpendicular to both (by the right-hand rule), proportional to the area they span. We can make this in geometric algebra as the dual of the 2-blade spanned by the vectors:

$$\mathbf{a} \times \mathbf{b} \equiv (\mathbf{a} \wedge \mathbf{b}) \cdot \widetilde{\mathbf{I}}_3. \tag{21}$$

This shows a number of things explicitly which one always needs to remember about the cross product: there is a convention involved on handedness (this is coded in the sign of $\mathbf{I}_3$); there are metric aspects since it is perpendicular to a plane (this is coded in the usage of the inner product '$\cdot$'); and the construction really only works in three dimensions, since only then is the dual of a 2-blade a vector (this is coded in the 3-gradedness of $\mathbf{I}_3$). The vector relationship $\mathbf{a} \wedge \mathbf{b}$ does not depend on any of these embedding properties, yet characterizes the $(\mathbf{a}, \mathbf{b})$-plane just as well.

You may verify that computing eq.(21) explicitly using eq.(1) and eq.(16) indeed retrieves the usual expression:

$$\mathbf{a} \times \mathbf{b} = (a_2 b_3 - a_3 b_2)\, \mathbf{e}_1 + (a_3 b_1 - a_1 b_3)\, \mathbf{e}_2 + (a_1 b_2 - a_2 b_1)\, \mathbf{e}_3 \quad (22)$$

In geometric algebra, we have the possibility of replacing the cross product by more elementary constructions. In Section 7.3 we discuss the advantages of doing so.

4 Intersecting subspaces

So far, we can span subspaces and consider their containment and orthogonality. Geometric algebra also contains operations to determine the *union* and *intersection* of subspaces. These are the join and meet operations. Several notations exist for these in literature, causing some confusion. For this paper, we will simply use the set notations $\cup$ and $\cap$ to make the formulas more easily readable.[3]

4.1 Union of subspaces

The join of two subspaces is their smallest superspace, i.e. the smallest space containing them both. Representing the spaces by blades $\mathbf{A}$ and $\mathbf{B}$, the join is denoted $\mathbf{A} \cup \mathbf{B}$. If the subspaces of $\mathbf{A}$ and $\mathbf{B}$ are disjoint, their join is obviously proportional to $\mathbf{A} \wedge \mathbf{B}$. But a problem is that if $\mathbf{A}$ and $\mathbf{B}$ are not disjoint (which is precisely the case we are interested in), then $\mathbf{A} \cup \mathbf{B}$ contains an unknown scaling factor which is fundamentally unresolvable due to the reshapable nature of the blades discussed in Section 2.3 (see Figure 4; this ambiguity was also observed by [13][Stolfi]). Fortunately, it appears that in all geometrically relevant entities which we compute this scalar ambiguity cancels.

The join is a more complicated product of subspaces than the outer product and inner product; we can give no simple formula for the grade of the result (like eq.(3)), and it cannot be characterized by a list of algebraic computation rules. Although computation of the join may appear to require some optimization process, finding the *smallest* superspace can actually be done in virtually constant time.

[3]We should also say that there are some issues currently being resolved to make meet and join a properly embedded part of geometric algebra since they produces blades modulo a multiplicative scaling factor rather than actual blades. Most literature now uses them only in projective geometry, in which there is no problem.

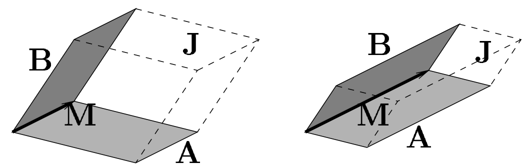


Figure 4: *The ambiguity of scale for* meet $\mathbf{M}$ *and* join $\mathbf{J}$ *of two blades* $\mathbf{A}$ *and* $\mathbf{B}$. *Both figures are examples of acceptable solutions.*

4.2 Intersection of subspaces

The meet of two subspaces $\mathbf{A}$ and $\mathbf{B}$ is their largest common subspace. If this is the blade $\mathbf{M}$, then $\mathbf{A}$ can be factorized as $\mathbf{A} = \mathbf{A}' \wedge \mathbf{M}$ and $\mathbf{B}$ as $\mathbf{B} = \mathbf{M} \wedge \mathbf{B}'$, and their join is a multiple of $\mathbf{A}' \wedge \mathbf{M} \wedge \mathbf{B}' = \mathbf{A} \wedge \mathbf{B}' = \mathbf{A}' \wedge \mathbf{B}$. This gives the relationship between meet and join.

Given the join $\mathbf{J} \equiv \mathbf{A} \cup \mathbf{B}$ of $\mathbf{A}$ and $\mathbf{B}$, we can compute their meet by the property that its dual (with respect to the join) is the outer product of their duals (this is a not-so-obvious consequence of the required 'containment in both'). In formula, this is:

$$(\mathbf{A} \cap \mathbf{B}) \cdot \widetilde{\mathbf{J}} = (\mathbf{B} \cdot \widetilde{\mathbf{J}}) \wedge (\mathbf{A} \cdot \widetilde{\mathbf{J}}) \ \text{ or } \ (\mathbf{A} \cap \mathbf{B})^* = \mathbf{B}^* \wedge \mathbf{A}^*$$

with the dual taken with respect to the join $\mathbf{J}$. (The somewhat strange order is a consequence of the factorization chosen above, and it corresponds to [13] for vectors). This leads to a formula for the meet of $\mathbf{A}$ and $\mathbf{B}$ relative to the chosen join (use eq.(18)) :

$$\mathbf{A} \cap \mathbf{B} = (\mathbf{B} \cdot \widetilde{\mathbf{J}}) \cdot \mathbf{A}. \tag{23}$$

Let us do an example: the intersection of two planes represented by the 2-blades $\mathbf{A} = \frac{1}{2}(\mathbf{e}_1 + \mathbf{e}_2) \wedge (\mathbf{e}_2 + \mathbf{e}_3)$ and $\mathbf{B} = \mathbf{e}_1 \wedge \mathbf{e}_2$. Note that we have normalized them (this is not necessary, but convenient for a point we want to make later). These are planes in general position in 3-dimensional space, so their join is proportional to $\mathbf{I}_3$. It makes sense to take $\mathbf{J} = \mathbf{I}_3$. This gives for the meet:

$$\begin{aligned} \mathbf{A} \cap \mathbf{B} &= \tfrac{1}{2}\left((\mathbf{e}_1 \wedge \mathbf{e}_2) \cdot (\mathbf{e}_3 \wedge \mathbf{e}_2 \wedge \mathbf{e}_1)\right) \cdot \left((\mathbf{e}_1 + \mathbf{e}_2) \wedge (\mathbf{e}_2 + \mathbf{e}_3)\right) \\ &= \tfrac{1}{2}\mathbf{e}_3 \cdot \left((\mathbf{e}_1 + \mathbf{e}_2) \wedge \mathbf{e}_3\right) \\ &= -\tfrac{1}{2}(\mathbf{e}_1 + \mathbf{e}_2) = -\frac{1}{\sqrt{2}}\left(\frac{\mathbf{e}_1 + \mathbf{e}_2}{\sqrt{2}}\right) \end{aligned} \tag{24}$$

(the last step expresses the result in normalized form). Figure 5 shows the answer; as in [13] the sign of $\mathbf{A} \cap \mathbf{B}$ is the right-hand rule applied to the turn required to make $\mathbf{A}$ coincide with $\mathbf{B}$, in the correct orientation.

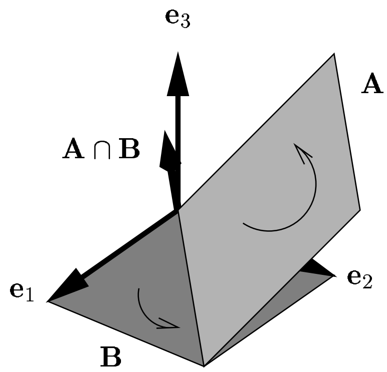


Figure 5: *An example of the* meet

Classically, one computes the intersection of two planes in 3-space by first converting them to normal vectors, and then taking the cross product. We can see that this gives the same answer in this non-degenerate case in 3-space, using our previous equations eq.(17), eq.(18), and noting that $\widetilde{\mathbf{I}}_3 = -\mathbf{I}_3$:

$$\begin{aligned}
(\mathbf{A} \cdot \widetilde{\mathbf{I}}_3) \times (\mathbf{B} \cdot \widetilde{\mathbf{I}}_3) &= \Big((\mathbf{A} \cdot \widetilde{\mathbf{I}}_3) \wedge (\mathbf{B} \cdot \widetilde{\mathbf{I}}_3)\Big) \cdot \widetilde{\mathbf{I}}_3 \\
&= \Big((\mathbf{B} \cdot \widetilde{\mathbf{I}}_3) \wedge (\mathbf{A} \cdot \widetilde{\mathbf{I}}_3)\Big) \cdot \mathbf{I}_3 \\
&= (\mathbf{B} \cdot \widetilde{\mathbf{I}}_3) \cdot \Big((\mathbf{A} \cdot \widetilde{\mathbf{I}}_3) \cdot \mathbf{I}_3\Big) \\
&= (\mathbf{B} \cdot \widetilde{\mathbf{I}}_3) \cdot \Big(\mathbf{A} \wedge (\widetilde{\mathbf{I}}_3 \cdot \mathbf{I}_3)\Big) \\
&= (\mathbf{B} \cdot \widetilde{\mathbf{I}}_3) \cdot \mathbf{A}.
\end{aligned}$$

So the classical result is a special case of eq.(23), but that formula is much more general: it applies to the intersection of subspaces of *any* grade, within a space of *any* dimension. With it, we begin to see some of the potential power of geometric algebra.

When the meet is a scalar, the two subspaces intersect in the point at the origin. This is in agreement with our geometrical interpretation in Section 2.7 of scalars as the weighted point at the origin. Scalars are geometrical objects, too!

The norm of the meet gives an impression of the 'strength' of the intersection. Between normalized subspaces in Euclidean space, the magnitude of the meet is the sine of the angle between them. From numerical analysis, this is a well-known measure for the 'distance' between subspaces in terms of their orthogonality: it is 1 if the spaces are orthogonal, and decays gracefully to 0 as the spaces get more parallel, before changing sign. This numerical significance is very useful in appli-

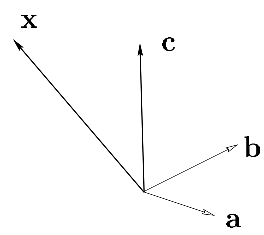


Figure 6: *Ratios of vectors*

cations.

5 Ratios of subspaces

With subspaces as basic elements of computation, we would really like to complete our algebra by the ability to solve equations in similarity problems such as indicated in Figure 6:

> Given two vectors $\mathbf{a}$ and $\mathbf{b}$, and a third vector $\mathbf{c}$, determine $\mathbf{x}$ so that $\mathbf{x}$ is to $\mathbf{c}$ as $\mathbf{b}$ is to $\mathbf{a}$, i.e. solve (in a symbolic notation which we will soon make exact):
> $$\frac{\mathbf{x}}{\mathbf{c}} = \frac{\mathbf{b}}{\mathbf{a}} \tag{25}$$

Such equations require a *division* of subspaces (here vectors), and so, really, an invertible product of subspaces. This *geometric product* is at the core of geometric algebra, and it is a rather amazing construction, at first sight.

5.1 The geometric product

For vectors, the geometric product is defined in terms of the inner and outer product as:

$$\mathbf{a}\,\mathbf{b} \equiv \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b} \tag{26}$$

So the geometric product of two vectors is an element of mixed grade: it has a scalar (0-blade) part $\mathbf{a} \cdot \mathbf{b}$ and a 2-blade part $\mathbf{a} \wedge \mathbf{b}$. It is therefore *not* a blade; rather, it is an operator on blades (as we will soon show). Changing the order of $\mathbf{a}$ and $\mathbf{b}$ gives:

$$\mathbf{b}\,\mathbf{a} \equiv \mathbf{b} \cdot \mathbf{a} + \mathbf{b} \wedge \mathbf{a} = \mathbf{a} \cdot \mathbf{b} - \mathbf{a} \wedge \mathbf{b}$$

The geometric product of two vectors is therefore neither fully symmetric (or rather: commutative), nor fully anti-symmetric.

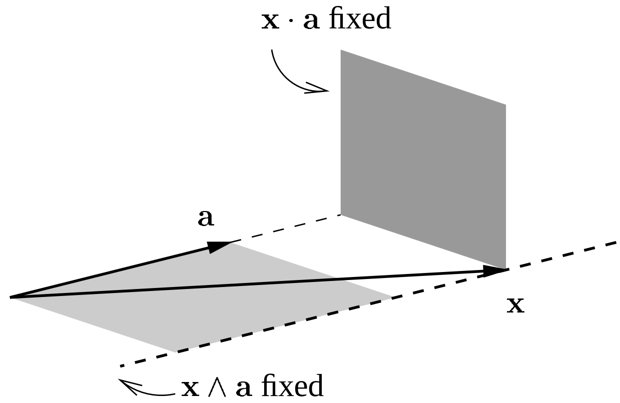


Figure 7: *Invertibility of the geometric products.*

A simple drawing may convince you that the geometric product is indeed invertible, whereas the inner and outer product separately are not. In Figure 7, we have a given vector $\mathbf{a}$. We denote the set of vectors $\mathbf{x}$ with the same value of the inner product $\mathbf{x} \cdot \mathbf{a}$ – this is a plane perpendicular to $\mathbf{a}$. The set of all vectors with the same value of the outer product $\mathbf{x} \wedge \mathbf{a}$ is also denoted – this is the line of all points which span the same directed area with $\mathbf{a}$. Neither of these sets is a singleton (in spaces of more than 1 dimension), so the inner and outer products are not fully invertible. The geometric product provides both the plane and the line, and therefore permits determining their unique intersection $\mathbf{x}$, as illustrated in the figure. Therefore it is invertible.

Note that the geometric product is sensitive to the relative directions of the vectors: for parallel vectors $\mathbf{a}$ and $\mathbf{b}$, the outer product contribution is zero, and $\mathbf{a}\,\mathbf{b}$ is a scalar and commutative in its factors; for perpendicular vectors, $\mathbf{a}\,\mathbf{b}$ is a 2-blade, and anti-commutative. In general, if the angle between $\mathbf{a}$ and $\mathbf{b}$ is ϕ in their common plane with unit 2-blade $\mathbf{I}$, we can write (in a Euclidean space):

$$\mathbf{a}\,\mathbf{b} = |\mathbf{a}|\,|\mathbf{b}|\,(\cos\phi + \mathbf{I}\,\sin\phi) \tag{27}$$

We will see below that $\mathbf{I}\,\mathbf{I} = -1$, so this is very reminiscent of complex numbers. More about that later, we mention it here to make the construction of the different grade elements in eq.(26) somewhat less outrageous than it may appear at first.

Eq.(26) defines the geometric product *only for vectors*. For arbitrary elements of our algebra it is defined using linearity and associativity, and making it coincide with the usual scalar product in the vector space, as the notation already suggests. That gives the following axioms (where α and β are scalars, $\mathbf{x}$ is a vector, A is a general element of the algebra):

$$\begin{array}{rll}
\text{scalars} & \alpha\,\beta \;\text{ and }\; \alpha\,\mathbf{x} \;\text{ have their usual meaning in } V^m & (28)\\
\text{scalars commute} & \alpha\,A = A\,\alpha & (29)\\
\text{vectors} & \mathbf{x}\,A = \mathbf{x}\cdot A + \mathbf{x}\wedge A & (30)\\
\text{associativity} & A\,(B\,C) = (A\,B)\,C & (31)\\
\text{distributivity 1} & A\,(B+C) = A\,B + A\,C & (32)\\
\text{distributivity 2} & (A+B)\,C = A\,C + B\,C & (33)
\end{array}$$

(One can avoid the reference to the inner and outer product through replacing eq.(30) by 'the square of a vector $\mathbf{x}$ must be equal to the scalar $Q(\mathbf{x},\mathbf{x})$', with Q the bilinear form of the vector space. Then one can re-introduce inner and outer product through the commutative properties of the geometric product:

$$\mathbf{a}\cdot\mathbf{b} = \tfrac{1}{2}(\mathbf{a}\,\mathbf{b} + \mathbf{b}\,\mathbf{a}) \;\text{ and }\; \mathbf{a}\wedge\mathbf{b} = \tfrac{1}{2}(\mathbf{a}\,\mathbf{b} - \mathbf{b}\,\mathbf{a}). \tag{34}$$

This is mathematically cleaner, but too indirect for our purpose here.)

It may not be obvious that these equations give enough information to compute the geometric product of arbitrary elements. Rather than show this abstractly, let us show by example how the rules can be used to develop the geometric algebra of 3-dimensional Euclidean space. We introduce, for convenience only, an orthonormal basis $\{\mathbf{e}_i\}_{i=1}^3$. Since this implies that $\mathbf{e}_i\cdot\mathbf{e}_j = \delta_{ij}$, we get the commutation rules:

$$\mathbf{e}_i\mathbf{e}_j = \begin{cases} -\mathbf{e}_j\mathbf{e}_i & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{35}$$

In fact, the former is equal to $\mathbf{e}_i\wedge\mathbf{e}_j$, whereas the latter equals $\mathbf{e}_i\cdot\mathbf{e}_i$. Considering the unit 2-blade $\mathbf{e}_1\wedge\mathbf{e}_2$, we find for its square:

$$\begin{aligned}
(\mathbf{e}_i\wedge\mathbf{e}_j)^2 &= (\mathbf{e}_i\wedge\mathbf{e}_j)\,(\mathbf{e}_i\wedge\mathbf{e}_j) = (\mathbf{e}_i\,\mathbf{e}_j)\,(\mathbf{e}_i\,\mathbf{e}_j)\\
&= \mathbf{e}_i\,\mathbf{e}_j\,\mathbf{e}_i\,\mathbf{e}_j = -\mathbf{e}_i\,\mathbf{e}_i\,\mathbf{e}_j\,\mathbf{e}_j = -1
\end{aligned} \tag{36}$$

So a unit 2-blade squares to -1 (we just computed for $\mathbf{e}_1\wedge\mathbf{e}_2$ for convenience, but there is nothing exceptional about that particular unit 2-blade, since the basis was arbitrary). Continued application of eq.(35) gives the full multiplication for all basis elements in the Clifford algebra of 3-dimensional space. The resulting multiplication table is given in Figure 8. Arbitrary elements are expressible as a linear combination of these basis elements, so this table determines the full algebra.

$\mathcal{C}\ell_3$	1	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_{12}$	$\mathbf{e}_{31}$	$\mathbf{e}_{23}$	$\mathbf{e}_{123}$
1	1	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_{12}$	$\mathbf{e}_{31}$	$\mathbf{e}_{23}$	$\mathbf{e}_{123}$
$\mathbf{e}_1$	$\mathbf{e}_1$	1	$\mathbf{e}_{12}$	$-\mathbf{e}_{31}$	$\mathbf{e}_2$	$-\mathbf{e}_3$	$\mathbf{e}_{123}$	$\mathbf{e}_{23}$
$\mathbf{e}_2$	$\mathbf{e}_2$	$-\mathbf{e}_{12}$	1	$\mathbf{e}_{23}$	$-\mathbf{e}_1$	$\mathbf{e}_{123}$	$\mathbf{e}_3$	$\mathbf{e}_{31}$
$\mathbf{e}_3$	$\mathbf{e}_3$	$\mathbf{e}_{31}$	$-\mathbf{e}_{23}$	1	$\mathbf{e}_{123}$	$\mathbf{e}_1$	$-\mathbf{e}_2$	$\mathbf{e}_{12}$
$\mathbf{e}_{12}$	$\mathbf{e}_{12}$	$-\mathbf{e}_2$	$\mathbf{e}_1$	$\mathbf{e}_{123}$	-1	$\mathbf{e}_{23}$	$-\mathbf{e}_{31}$	$-\mathbf{e}_3$
$\mathbf{e}_{31}$	$\mathbf{e}_{31}$	$\mathbf{e}_3$	$\mathbf{e}_{123}$	$-\mathbf{e}_1$	$-\mathbf{e}_{23}$	-1	$\mathbf{e}_{12}$	$-\mathbf{e}_2$
$\mathbf{e}_{23}$	$\mathbf{e}_{23}$	$\mathbf{e}_{123}$	$-\mathbf{e}_3$	$\mathbf{e}_2$	$\mathbf{e}_{31}$	$-\mathbf{e}_{12}$	-1	$-\mathbf{e}_1$
$\mathbf{e}_{123}$	$\mathbf{e}_{123}$	$\mathbf{e}_{23}$	$\mathbf{e}_{31}$	$\mathbf{e}_{12}$	$-\mathbf{e}_3$	$-\mathbf{e}_2$	$-\mathbf{e}_1$	-1

Figure 8: The multiplication table of the geometric algebra of 3-dimensional Euclidean space, on an orthonormal basis. Shorthand: $\mathbf{e}_{12} = \mathbf{e}_1 \wedge \mathbf{e}_2$, etcetera.

5.2 Invertibility of the geometric product

The geometric product is invertible, so 'dividing by a vector' has a unique meaning. We will usually do this through 'multiplication by the inverse of the vector'. Since multiplication is not necessarily commutative, we have to be a bit careful: there is a 'left division' and a 'right division'.

As you may verify, the unique inverse of a vector $\mathbf{a}$ is:

$$\mathbf{a}^{-1} = \frac{\mathbf{a}}{\mathbf{a} \cdot \mathbf{a}} = \frac{\mathbf{a}}{|\mathbf{a}|^2}$$

since that is the unique element that satisfies: $\mathbf{a}^{-1}\,\mathbf{a} = 1 = \mathbf{a}\,\mathbf{a}^{-1}$. Similarly, a blade $\mathbf{A}$ (of which the norm should not be zero) has the inverse

$$\mathbf{A}^{-1} = \frac{\widetilde{\mathbf{A}}}{\mathbf{A} \cdot \widetilde{\mathbf{A}}} = \frac{\widetilde{\mathbf{A}}}{|\mathbf{A}|^2}$$

(the reverse is due to the definition of the norm in eq.(6)).

5.3 Projection of subspaces

The availability of an inverse gives us an interesting of way of decomposing a vector $\mathbf{x}$ relative to a given blade $\mathbf{A}$ using the geometric product:

$$\mathbf{x} = (\mathbf{x}\,\mathbf{A})\,\mathbf{A}^{-1} = (\mathbf{x} \cdot \mathbf{A})\,\mathbf{A}^{-1} + (\mathbf{x} \wedge \mathbf{A})\,\mathbf{A}^{-1} \tag{37}$$

The first term is a blade fully inside $\mathbf{A}$: it is the *projection* of $\mathbf{x}$ onto $\mathbf{A}$. The second term is a vector perpendicular to $\mathbf{A}$, sometimes called the *rejection* of $\mathbf{x}$ by $\mathbf{A}$. The

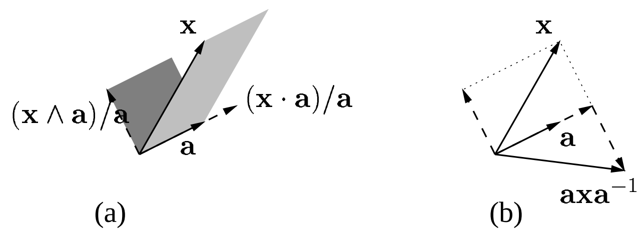


Figure 9: *(a) Projection and rejection of* $\mathbf{x}$ *relative to* $\mathbf{a}$*. (b) Reflection of* $\mathbf{x}$ *in* $\mathbf{a}$*.*

projection of a blade $\mathbf{X}$ onto a blade $\mathbf{A}$ is given by the extension of the above, as:

$$\text{projection of } \mathbf{X} \text{ onto } \mathbf{A}: \quad \mathbf{X} \;\mapsto\; (\mathbf{X} \cdot \mathbf{A})\, \mathbf{A}^{-1}$$

Again geometric algebra has allowed a straightforward extension to arbitrary dimensions of subspaces, without additional computational complexity.

5.4 Reflection of subspaces

The *reflection* of a vector $\mathbf{x}$ relative to a fixed vector $\mathbf{a}$ can be constructed from the decomposition of eq.(37) (used for a vector $\mathbf{a}$), by changing the sign of the rejection (see Figure 9b). This can be rewritten in terms of the geometric product:

$$(\mathbf{x} \cdot \mathbf{a})\, \mathbf{a}^{-1} - (\mathbf{x} \wedge \mathbf{a})\, \mathbf{a}^{-1} = (\mathbf{a} \cdot \mathbf{x} + \mathbf{a} \wedge \mathbf{x})\, \mathbf{a}^{-1} = \mathbf{a}\, \mathbf{x}\, \mathbf{a}^{-1}.$$

So the reflection of $\mathbf{x}$ in $\mathbf{a}$ is the expression $\mathbf{a}\,\mathbf{x}\,\mathbf{a}^{-1}$, see Figure 9b; the reflection in a plane perpendicular to $\mathbf{a}$ is then $-\mathbf{a}\,\mathbf{x}\,\mathbf{a}^{-1}$,

We can extend this formula to the reflection of a blade $\mathbf{X}$ relative to the vector $\mathbf{a}$, this is simply:

$$\text{reflection in vector } \mathbf{a}: \quad \mathbf{X} \;\mapsto\; \mathbf{a}\, \mathbf{X}\, \mathbf{a}^{-1}.$$

and even to the reflection of a blade $\mathbf{X}$ in a k-blade $\mathbf{A}$, which turns out to be:

$$\text{general reflection:} \quad \mathbf{X} \;\mapsto\; -(-1)^k\, \mathbf{A}\, \mathbf{X}\, \mathbf{A}^{-1}.$$

Note that these formulas permit you to do reflections of subspaces without first decomposing them in constituent vectors. It gives the possibility of reflection a polyhedral object by directly using a facet representation, rather than acting on individual vertices.

5.5 Angles as geometrical objects

We have found in eq.(36) that any unit 2-blade $\mathbf{I}$ in a Euclidean space satisfies $\mathbf{I}^2 = -1$, so this is also true for the unit 2-blade occurring in eq.(27). Therefore, using the usual definition of the exponential as a converging series of terms, we are actually permitted to write the geometric product in an exponential form:

$$\mathbf{a}\,\mathbf{b} = |\mathbf{a}|\,|\mathbf{b}|\,(\cos\phi + \mathbf{I}\,\sin\phi) = |\mathbf{a}|\,|\mathbf{b}|\,e^{\mathbf{I}\phi} \tag{38}$$

with $\mathbf{I}$ the unit 2-blade containing $\mathbf{a}$ and $\mathbf{b}$, oriented from $\mathbf{a}$ to $\mathbf{b}$. This exponential form will be very convenient when we do rotations. Note that all elements occurring in this equation have a straightforward geometrical interpretation, we are not doing complex numbers here! (Really, we aren't: $\mathbf{I}$ is not a complex scalar, since then it would have to commute with *all* elements of the algebra by eq.(29), but it instead satisfies $\mathbf{a}\,\mathbf{I} = -\mathbf{I}\mathbf{a}$ for vectors $\mathbf{a}$ in the $\mathbf{I}$-plane.)

The combination $\mathbf{I}\phi$ is a full indication of the angle between the two vectors: it denotes not only the magnitude, but also the plane in which the angle is measured, and even the orientation of the angle. If you ask for the scalar magnitude of the geometrical quantity $\mathbf{I}\phi$ in the plane $-\mathbf{I}$ (the plane 'from $\mathbf{b}$ to $\mathbf{a}$' rather than 'from $\mathbf{a}$ to $\mathbf{b}$'), it is $-\phi$; so the scalar value of the angle automatically gets the right sign. The fact that the angle as expressed by $\mathbf{I}\phi$ is now a geometrical quantity independent of the convention used in its definition removes a major headache from many geometrical computations involving angles. We call this true geometric quantity the *bivector angle* (it is just a 2-blade, of course, not a new kind of element – but we use it as an angle, hence the name).

5.6 Rotations in the plane

Using the inverse of a vector, we can now solve the motivating problem of eq.(25), to find a vector $\mathbf{x}$ that is to $\mathbf{c}$ as $\mathbf{b}$ is to $\mathbf{a}$. Denoting the 2-blade of the $(\mathbf{a} \wedge \mathbf{b})$-plane by $\mathbf{I}$, we obtain:

$$\mathbf{x}\,\mathbf{c}^{-1} = \mathbf{b}\,\mathbf{a}^{-1}$$

so that

$$\mathbf{x} = (\mathbf{b}\,\mathbf{a}^{-1})\,\mathbf{c} = \frac{|\mathbf{b}|}{|\mathbf{a}|}e^{-\mathbf{I}\phi}\mathbf{c} \tag{39}$$

Here $\mathbf{I}\phi$ is the angle in the $\mathbf{I}$ plane from $\mathbf{a}$ to $\mathbf{b}$, as in eq.(38), so $-\mathbf{I}\phi$ is the angle from $\mathbf{b}$ to $\mathbf{a}$. If we happen to have $|\mathbf{a}| = |\mathbf{b}|$, we get $\mathbf{x} = e^{-\mathbf{I}\phi}\mathbf{c}$; apparently we should interpret 'pre-multiplying by $e^{-\mathbf{I}\phi}$' as a *rotation operator* in the $\mathbf{I}$-plane. The full expression of eq.(39) denotes a rotation/dilation in the $\mathbf{I}$-plane.

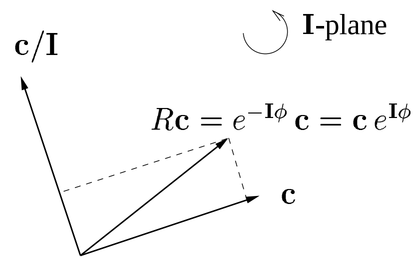


Figure 10: *Coordinate-free specification of rotation.*

Let us write this out, to get familiar with the geometric algebra way of looking at rotations:

$$e^{-\mathbf{I}\phi}\,\mathbf{c} = \mathbf{c}\cos\phi - \mathbf{I}\mathbf{c}\sin\phi = \mathbf{c}\cos\phi + \mathbf{c}\mathbf{I}\sin\phi$$

What is $\mathbf{c}\mathbf{I}$? Introduce orthonormal coordinates $\{\mathbf{e}_1, \mathbf{e}_2\}$ in the $\mathbf{I}$-plane, with $\mathbf{e}_1$ along $\mathbf{c}$, so that $\mathbf{c} \equiv c\,\mathbf{e}_1$. Then $\mathbf{I} = \mathbf{e}_1 \wedge \mathbf{e}_2 = \mathbf{e}_1\,\mathbf{e}_2$. Therefore $\mathbf{c}\mathbf{I} = c\,\mathbf{e}_1\,\mathbf{e}_1\mathbf{e}_2 = c\,\mathbf{e}_2$: it is $\mathbf{c}$ *turned over a right angle*, following the orientation of the 2-blade $\mathbf{I}$ (here anti-clockwise). So $\mathbf{c}\cos\phi + \mathbf{c}\mathbf{I}\sin\phi$ is 'a bit of $\mathbf{c}$ plus a bit of its anti-clockwise perpendicular' – and those amounts are precisely right to make it equal to the rotation by ϕ, see Figure 10.

If you use a classical rotation matrix in 2 dimensions, it does precisely this construction, but in a coordinate system that is adapted to an arbitrary basis $\{\mathbf{e}_1, \mathbf{e}_2\}$, rather than to $\mathbf{c}$. That is why you then need 4 coefficients, to describe how each of those 2 basis vectors turns. Geometric algebra is coordinate-free in this sense: orthogonal directions can be made from the vectors for which you need them in a coordinate-free manner. Then a specification of the rotation requires only 2 trigonometric functions, just for the scaling of those 2 components.

5.7 Rotations in 3 dimensions

Two subsequent reflections in lines which make an angle of $\phi/2$ in a plane with unit 2-blade $\mathbf{I}$ constitute a rotation over ϕ in the $\mathbf{I}$-plane. In 2-dimensional space, this is obvious, but it also works in 3-dimensional space, see Figure 11 (and even in m-dimensional space). It gives us the way to express general rotations in geometric algebra.

Two successive reflections of a vector $\mathbf{x}$ in vectors $\mathbf{u}$ and $\mathbf{v}$ give

$$\mathbf{v}\,(\mathbf{u}\,\mathbf{x}\,\mathbf{u}^{-1})\,\mathbf{v}^{-1} = \frac{\mathbf{v}}{|\mathbf{v}|}\,\frac{\mathbf{u}}{|\mathbf{u}|}\,\mathbf{x}\,\frac{\mathbf{u}}{|\mathbf{u}|}\,\frac{\mathbf{v}}{|\mathbf{v}|} = e^{-\mathbf{I}\phi/2}\,\mathbf{x}\,e^{\mathbf{I}\phi/2}$$

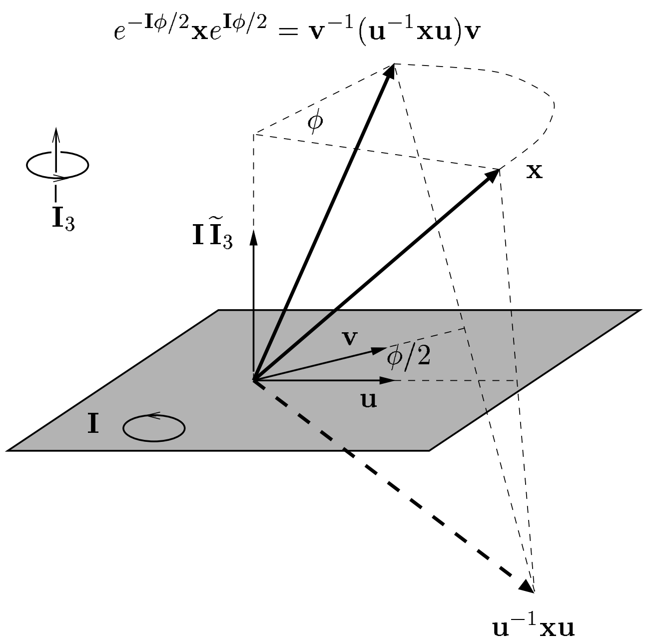


Figure 11: *A rotation as 2 reflections in vectors* $\mathbf{u}$ *and* $\mathbf{v}$*, making an angle of* $\mathbf{I}\phi/2$*.*

where we used the exponential notation for the geometric product of two unit vectors ($\mathbf{I}$ is the unit 2-blade from $\mathbf{u}$ to $\mathbf{v}$). The expression for the rotation is therefore directly given by the bivector angle, i.e. by angle and rotation plane. An operator $e^{-\mathbf{I}\phi/2}$, used in this way, is called a *rotor*. Writing out this expression in terms of the perpendicular component $\mathbf{x}_\perp$ (rejection) and the parallel component $\mathbf{x}_{||}$ (projection) of $\mathbf{x}$ relative to the $\mathbf{I}$ plane gives

$$\text{rotation over } \mathbf{I}\phi\text{:} \quad \mathbf{x} \;\mapsto\; e^{-\mathbf{I}\phi/2}\,\mathbf{x}\,e^{\mathbf{I}\phi/2} = \mathbf{x}_\perp + e^{-\mathbf{I}\phi}\,\mathbf{x}_{||} \tag{40}$$

(this is a good exercise, it requires $\mathbf{I}\,\mathbf{x}_\perp = \mathbf{x}_\perp\,\mathbf{I}$ and $\mathbf{I}\,\mathbf{x}_{||} = -\mathbf{x}_{||}\,\mathbf{I}$; why do these hold?). So the perpendicular component to the rotation plane is unchanged (as it should!), and the parallel component becomes pre-multiplied by $e^{-\mathbf{I}\phi}$. We have seen in eq.(39) that this is a rotation in the $\mathbf{I}$-plane. (In fact, we could have defined the higher dimensional rotation by the right hand side of eq.(40) and then derived the left hand side.)

5.8 Combining rotations

Two successive rotations R_1 and R_2 are equivalent to a single new rotation R of which the rotor R is the geometric product of the rotors R_1 and R_2, since

$$R_2\, R_1\, \mathbf{x}\, R_1^{-1}\, R_2^{-1} = (R_2\, R_1)\, \mathbf{x}\, (R_2\, R_1)^{-1} \equiv R\, \mathbf{x} R^{-1}.$$

This applies in 3-dimensional space as well as in 2-dimensional space. Therefore the combination of rotations is a simple consequence of the definition of the geometric product on rotors, i.e. elements of the form $e^{-\mathbf{I}\phi/2} = \cos\phi/2 - \mathbf{I}\sin\phi/2$, with $\mathbf{I}^2 = -1$. (We could allow a scalar factor in the rotor, since the inverse divides it out; yet it is common to restrict rotor to be normalized to unity – then one can replace R^{-1} by $\widetilde{R}$, defining the rotation by $R\,\mathbf{x}\,\widetilde{R}$. Reversion is a simpler (cheaper) operation than inversion, though the normalization may add some additional computational cost.)

Let's see how it works in 3-space. In 3 dimensions, we are used to specifying rotations by a *rotation axis* $\mathbf{a}$ rather than by a *rotation plane* $\mathbf{I}$. The relationship between axis and plane is given by duality: $\mathbf{a} \equiv \mathbf{I} \cdot \widetilde{\mathbf{I}}_3 = -\mathbf{I}\,\mathbf{I}_3$ (check that this indeed gives the correct orientation). Given the axis $\mathbf{a}$, we therefore find the plane as the 2-blade $\mathbf{I} = -\mathbf{a}\,\mathbf{I}_3^{-1} = \mathbf{a}\mathbf{I}_3 = \mathbf{I}_3\mathbf{a}$. A rotation over an angle ϕ around an axis with unit vector $\mathbf{a}$ is therefore represented by the rotor $e^{-\mathbf{I}_3\mathbf{a}\phi/2}$.

To compose, say, a rotation R_1 around the $\mathbf{e}_1$ axis of $\pi/2$ with a subsequent rotation R_2 over the $\mathbf{e}_2$ axis over $\pi/2$, we write out their rotors:

$$R_1 = e^{-\mathbf{I}_3\mathbf{e}_1\pi/4} = \frac{1-\mathbf{e}_{23}}{\sqrt{2}} \quad \text{and} \quad R_2 = e^{-\mathbf{I}_3\mathbf{e}_2\pi/4} = \frac{1-\mathbf{e}_{31}}{\sqrt{2}}$$

The total rotor is their product, and we rewrite it back to the exponential form to find the axis:

$$\begin{aligned} R \equiv R_2\, R_1 &= \tfrac{1}{2}(1-\mathbf{e}_{23})\,(1-\mathbf{e}_{31}) = \tfrac{1}{2}(1-\mathbf{e}_{23}-\mathbf{e}_{31}-\mathbf{e}_{12}) \\ &= \tfrac{1}{2} - \tfrac{1}{2}\sqrt{3}\,\mathbf{I}_3\,\frac{\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_3}{\sqrt{3}} = e^{-\mathbf{I}_3\mathbf{a}\pi/3} \end{aligned}$$

Therefore the total rotation is over the axis $\mathbf{a} = (\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_3)/\sqrt{3}$, over the angle $2\pi/3$. But of course you do not need to decompose the resulting rotor into those geometrical constituents: you can apply it immediately to a vector $\mathbf{x}$ as $R\,\mathbf{x}\,R^{-1}$, or even to an arbitrary blade through the formula:

$$\text{general rotation:}\quad \mathbf{X} \;\mapsto\; R\,\mathbf{X}\,R^{-1}$$

This enables you to rotate a plane in one operation, for instance:

$$R(\mathbf{e}_1 \wedge \mathbf{e}_2)R^{-1} \;=\; \tfrac{1}{4}(1-\mathbf{e}_{23}-\mathbf{e}_{31}-\mathbf{e}_{12})\,\mathbf{e}_{12}\,(1+\mathbf{e}_{23}+\mathbf{e}_{31}+\mathbf{e}_{12}) = \mathbf{e}_{23}$$

No need to decompose the plane into its spanning vectors first!

5.9 Quaternions: based on bivectors

You may have recognized the example above as strongly similar to quaternion computations. Quaternions are indeed part of geometric algebra, in the following straightforward manner.

Choose an orthonormal basis $\{\mathbf{e}_i\}_{i=1}^3$. Construct out of that a bivector basis with elements $\mathbf{e}_{12} \equiv \mathbf{e}_1 \wedge \mathbf{e}_2 (= \mathbf{e}_1\, \mathbf{e}_2)$ and cyclic. Note that these elements satisfy: $\mathbf{e}_{12}^2 = \mathbf{e}_{23}^2 = \mathbf{e}_{31}^2 = -1$, and $\mathbf{e}_{12}\, \mathbf{e}_{23} = \mathbf{e}_{13}$ (and cyclic) and also $\mathbf{e}_{12}\, \mathbf{e}_{23}\, \mathbf{e}_{31} = 1$. In fact, setting $i \equiv \mathbf{e}_{23}$, $j \equiv -\mathbf{e}_{31}$ and $k \equiv \mathbf{e}_{12}$, we find $i^2 = j^2 = k^2 = i\, j\, k = -1$ and $j\, i = k$ and cyclic. Algebraically these objects are the quaternions obeying the quaternion product, commonly interpreted as some kind of '4-D complex number system'. There is nothing 'complex' about quaternions; but they are not really vectors either (as some still think) – they are just real 2-blades in 3-space, denoting elementary rotation planes, and multiplying through the geometric product. Visualizing quaternions is therefore straightforward: each is just a rotation plane with a rotation angle, and the 'bivector angle' concept represents that well (the corresponding quaternion is simply its exponential, elevating the bivector angle to a rotation operator).

5.10 Constructing rotors

For a 2-dimensional rotation, if you know for certain that a vector $\mathbf{e}$ has been rotated to become a vector $\mathbf{f}$ (which therefore necessarily has the same norm) by a rotation in the $\mathbf{e} \wedge \mathbf{f}$-plane, it is easy to find a rotor that does that:

$$R = 1 + \mathbf{f}\mathbf{e}$$

(if you want the unit rotor, you need to normalize this). For a 3-dimensional rotation, if you know an orthonormal frame $\{\mathbf{e}_i\}_{i=1}^3$ which has rotated to the frame $\{\mathbf{f}_i\}_{i=1}^3$, then a rotor doing that is:

$$R = 1 + \mathbf{f}_1\mathbf{e}_1 + \mathbf{f}_2\mathbf{e}_2 + \mathbf{f}_3\mathbf{e}_3$$

(which needs to be normalized if you want a unit rotor). This formula can be generalized simply to non-orthonormal frames, see [11]. Warning: the formulas do not work for rotations over π (there is then no unique rotation plane!) – but are very useful elsewhere.

6 Differentiation

Geometric algebra also has a much extended operation of differentiation, which contains the classical vector calculus, and much more. It is possible to differentiate with respect to a scalar or a vector, as before, but now also with respect to k-blades. This enables efficient encoding of differential geometry, in a coordinate-free manner, and gives an alternative look at differential shape descriptors like the 'second fundamental form' (it becomes an immediate indication of how the tangent plane changes when we slide along the surface).

Somebody should rewrite classical differential geometry texts into geometric algebra; but this has not been done yet and it would lead too far to do so in this introductory paper. Let us just briefly show the scalar differentiation of a rotor, to demonstrate how the commutation rules of geometric algebra naturally group to a well-known classical result, which is then automatically extended beyond vectors.

So, suppose we have a rotor $R = e^{-\mathbf{I}\phi/2}$, and use it to produce a rotated version $\mathbf{X} = R\,\mathbf{X}_0\,\widetilde{R}$ of some constant blade $\mathbf{X}_0$. Scalar differentiation with respect to time gives (using chain rule and commutation rules):

$$
\begin{aligned}
\tfrac{d}{dt}\mathbf{X} &= \tfrac{d}{dt}(e^{-\mathbf{I}\phi/2}\mathbf{X}_0 e^{\mathbf{I}\phi/2}) \\
&= -\tfrac{1}{2}\tfrac{d}{dt}(\mathbf{I}\phi)(e^{-\mathbf{I}\phi/2}\mathbf{X}_0 e^{\mathbf{I}\phi/2}) + \tfrac{1}{2}(e^{-\mathbf{I}\phi/2}\mathbf{X}_0 e^{\mathbf{I}\phi/2})\tfrac{d}{dt}(\mathbf{I}\phi) \\
&= \tfrac{1}{2}(\mathbf{X}\,\tfrac{d}{dt}(\mathbf{I}\phi) - \tfrac{d}{dt}(\mathbf{I}\phi)\,\mathbf{X}) \\
&= \mathbf{X} \times \tfrac{d}{dt}(\mathbf{I}\phi)
\end{aligned}
$$

using the commutator product $\times$ defined in geometric algebra as the shorthand $A \times B \equiv \frac{1}{2}(AB - BA)$; this product often crops up in computations with Lie groups such as the rotations. This simple expression which results assumes a more familiar form when $\mathbf{X}$ is a vector $\mathbf{x}$ in 3-space, the rotation plane is fixed so that $\frac{d}{dt}\mathbf{I} = 0$, and we introduce a scalar angular velocity $\omega \equiv \frac{d}{dt}\phi$. It is then common practice to introduce the vector dual to the plane as the angular velocity vector $\boldsymbol{\omega}$, so $\boldsymbol{\omega} \equiv \omega\mathbf{I} \cdot \widetilde{\mathbf{I}}_3 = \omega\mathbf{I}/\mathbf{I}_3$. We then obtain:

$$\tfrac{d}{dt}\mathbf{x} = \mathbf{x} \cdot \tfrac{d}{dt}(\mathbf{I}\phi) = \mathbf{x} \cdot (\boldsymbol{\omega}\,\mathbf{I}_3) = (\mathbf{x} \wedge \boldsymbol{\omega})\,\mathbf{I}_3 = \boldsymbol{\omega} \boldsymbol{\times} \mathbf{x}$$

where $\boldsymbol{\times}$ is the vector cross product. As before when we treated the meet and other operations, we find that an equally simple geometric algebra expression is much more general; here it describes the differential rotation of k-dimensional subspaces in n-dimensional space, rather than merely of vectors in 3-D.

Similar generalizations result for differentiation relative to blades; the interested reader is referred to the tutorial of [2], which introduces these differentiations using examples from physics.

7 Linear algebra

In the classical ways of using vector spaces, linear algebra is an important tool. In geometric algebra, this remains true: linear transformations are of interest in their own right, or as first order approximations to more complicated mappings. Indeed, linear algebra is an integral part of geometric algebra, and acquires much extended coordinate-free methods through this inclusion. We show some of the basic principles; much more may be found in [2] or [10].

7.1 Outermorphisms: spanning is linear

When vectors are transformed by a linear transformation on the vector space, the blades they span can be viewed to transform as well, simply by the rule: 'the transform of a span of vectors is the span of the transformed vectors'. This means that a linear transformation $\mathsf{f} : V^n \to V^n$ on a vector space has a natural extension to the whole geometric algebra of that vector space, as an *outermorphism,* i.e. a mapping that preserves the outer product structure:

$$\mathsf{f}(\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \cdots \wedge \mathbf{a}_k) \equiv \mathsf{f}(\mathbf{a}_1) \wedge \mathsf{f}(\mathbf{a}_2) \wedge \cdots \wedge \mathsf{f}(\mathbf{a}_k).$$

Note that this is grade-preserving: a k-blade transforms to a k-blade. To this we have to add what the extension does to scalars, which is simply: $\mathsf{f}(\alpha) = \alpha$.

This outermorphism definition has immediate consequences. Apply it to a pseudoscalar $\mathbf{I}_m$, which is an m-blade: it must produce another m-blade. But the linear space of m-blades in m-dimensional vector space is 1-dimensional, so this must again be a multiple of $\mathbf{I}_m$. That multiple is precisely the determinant of f in m-dimensional space:

$$\det(\mathsf{f}) = \mathsf{f}(\mathbf{I}_m)\mathbf{I}_m^{-1}.$$

The determinant is thus simply the change of hypervolume under f. This is nothing new, but it is satisfying that all the usual properties of the determinant, including its expression in terms of coordinates, follow immediately from this straightforward, coordinate-free definition.

7.2 Linear transformation of the inner product

The transformation rule for the inner product now follows automatically from the definition through eq.(7), and is found to be rather more involved:

$$\mathsf{f}(A \cdot B) = \overline{\mathsf{f}}^{\,-1}(A) \cdot \mathsf{f}(B),$$

where $\overline{\mathsf{f}}$ is the *adjoint* of f, defined by

$$\mathsf{f}(A) * B = A * \overline{\mathsf{f}}(B) \quad \text{for all } A \text{ and } B.$$

(In terms of matrices on an orthonormal basis, $\overline{\mathsf{f}}$ is the mapping represented by the *transpose* of the matrix representing f.)

7.3 No normal vectors or cross products!

Since the inner product transformation under a linear mapping is so involved, one should steer clear of any constructions that involve the inner product, especially in the characterization of basic properties of one's objects. Therefore the practice of characterizing a plane by its normal vector – which contains the inner product in its duality, see Section 3.3 – should be avoided. Under linear transformations, *the normal vector of a transformed plane is not the transform of the normal vector of the plane!* (this is a well known fact, but always a shock to novices). The normal vector is in fact a cross product of vectors, which (as you may verify from eq.(21) and the above) transforms as:

$$\mathsf{f}(\mathbf{a} \times \mathbf{b}) = \bar{\mathsf{f}}^{-1}(\mathbf{a}) \times \bar{\mathsf{f}}^{-1}(\mathbf{b}) / \det(\mathsf{f})$$

and that is usually not equal to $\mathsf{f}(\mathbf{a}) \times \mathsf{f}(\mathbf{b})$. It is therefore much better to characterize the plane by a 2-blade, now that we can. *The 2-blade of the transformed plane is the transform of the 2-blade of the plane*, since linear transformations are outermorphisms preserving the 2-blade construction. Especially when the planes are tangent planes constructed by differentiation, 2-blades are appropriate: under *any* transformation f, the construction of the tangent plane is only dependent on the first order linear approximation mapping f of f. Therefore a tangent plane represented as a 2-blade transforms simply under *any* transformation (and the same applies of course to tangent k-blades in higher dimensions). Using blades for those tangent spaces should enormously simplify the treatment of object through differential geometry, especially in the context of affine transformations – but this has not yet been done.

8 All you need is blades: models of geometries

So far we have been treating only homogeneous subspaces of the vector spaces, i.e. subspaces containing the origin. We have spanned them, projected them, and rotated them, but we have not moved them out of the origin to make more interesting geometrical structures such as lines floating in space.

There is a very nice way of making such basic primitives in geometric algebra. At first it looks like a straightforward embedding of the classical ideas behind 'homogeneous coordinates', but it rapidly becomes much more powerful than that. It creates an *algebra of points* (rather than vectors). We present three models of Euclidean space, all useful to computer graphics, and show how the geometric algebra of those models implements totally different semantics using the same basic products (but in different spaces). This goes much beyond resolving the issues raised in the classical papers by Goldman [6, 7].

8.1 The vector space model

The most straightforward model of Euclidean space represents its points by the translation vectors required to get there. We call those *position vectors*. This representation strongly depends on the location of the origin. It is well known [6] that this easily leads to bad representations and software which depend heavily on the chosen origin. It is inappropriate to take the position vectors $\mathbf{a}$ and $\mathbf{b}$ as 'being' the points A and B, and then form new points by addition of their vectors. The construction $\mathbf{a} + \mathbf{b}$ cannot represent a geometrical point, for its value changes as the origin changes, and no geometrically relevant objects should depend on that.

Still, the vector space model of a Euclidean space is appropriate for translation vectors (the null translation *is* special: it is the identity operation) and for tangent planes to a manifold (again, the origin is special since it is where the tangent space is attached to the manifold). For those, $\mathbf{a}+\mathbf{b}$ has a clear meaning: it is the resultant translation or resultant velocity, of a point. Beyond these applications, one has to be careful with the vector space model.

The products between vectors are just as much part of the model as the embedding of the points themselves (this is a point which Goldman [6, 7] neglects somewhat in his discussion of representations). In the vector space model, they simply have the meaning we have used throughout this paper: the *outer product* constructs the higher-dimensional proper subspaces; the *inner product* constructs the orthogonal complement of subspaces; and the *geometric product* gives us the rotation/dilation operator between subspaces. Elementary combinations of these give us projection and reflection. Note that all these operations are origin-centered in this model: rotations are around an axis through the origin, reflections are in planes through the origin, etcetera. It is simple to shift them out of the origin of course, but algebraically, that is a 'hack' – it would be much more tidy if we could find a representation in which those operations are all elementary relationships between blades (and we will). Even an basic concept like the Euclidean distance between two points P and Q is a fairly involved expression – we have to form $\sqrt{(\mathbf{p}-\mathbf{q})\cdot(\mathbf{p}-\mathbf{q})}$ to obtain this geometric invariant. It would be much nicer if this elementary concept were one of the elementary products.

The vector space model, then, contains a lot of the basic elements to do Euclidean geometry, especially when we consider its full geometric algebra of higher dimensional subspaces. But we can do better, tidying up the algebra by embedding Euclidean geometry of E^m in a space of more than m dimensions and using the geometric algebra of that space to describe the Euclidean objects and operators of interest.

8.2 The homogeneous model

We can get rid of the special nature of the origin, by (paradoxically!) introducing a vector representing it. To represent an m-dimensional Euclidean space E^m in this way, we must introduce an extra dimension and obtain an $(m+1)$-dimensional representation space. This is the familiar *homogeneous model* or *affine model* of the vector space.

8.2.1 Points as vectors

Let the unit vector for the extra dimension be denoted by e_0. This vector must be perpendicular to all regular vectors in the Euclidean space E^m, so $e_0 \cdot \mathbf{x} = 0$ for all $\mathbf{x} \in E^m$. We let e_0 denote 'the point at the origin'. A point at any other location $\mathbf{p}$ is made by translation of the point at the origin over $\mathbf{p}$. This is done by adding $\mathbf{p}$ to e_0. This construction therefore gives the representation of the point at location $\mathbf{p}$ as the vector p in $(m+1)$-dimensional space:

$$p = e_0 + \mathbf{p}$$

This is no more than the usual homogeneous coordinates; we have extended the m-dimensional vector by an e_0-coordinate to make an $(m+1)$-dimensional vector capable of representing a point in m-dimensional space.

We will denote vectors in the m-dimensional Euclidean space in **bold**, and vectors in the $(m+1)$-dimensional model in *italic*. You can visualize this construction as in Figure 12a (necessarily drawn for $m = 2$).

8.2.2 Off-set flats as blades

Now let us look at how we can interpret the higher grade elements of the geometric algebra of this $(m+1)$-dimensional space. A vector in $(m+1)$-space is apparently the representation of a point in E^m, i.e. a 0-dimensional affine subspace element. What does a 2-blade $p \wedge q$ formed by two vectors p and q represent, in other words, what is the semantics of the outer product in this homogeneous model? We compute

$$p \wedge q = (e_0 + \mathbf{p}) \wedge (e_0 + \mathbf{q}) = e_0 \wedge (\mathbf{q} - \mathbf{p}) + \mathbf{p} \wedge \mathbf{q}$$

We recognize the vector $\mathbf{q} - \mathbf{p}$, and the area spanned by $\mathbf{p}$ and $\mathbf{q}$. Both are elements which we need to describe an element of the directed line through the points p and q. The former is the *direction vector* of the directed line, the latter is an area which we will call the *moment* of the line through p and q. It denotes the distance to the origin, for we can rewrite it to a rectangle spanned by the direction $(\mathbf{q} - \mathbf{p})$ and

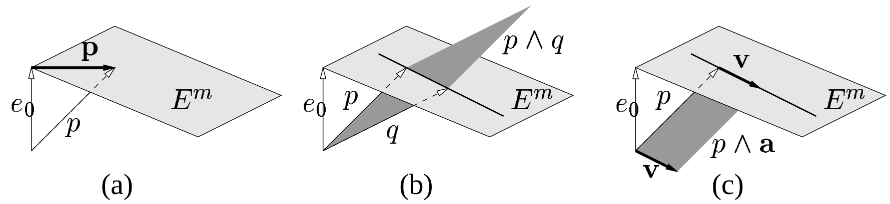


Figure 12: *Representing offset subspaces of* E^m *in* $m+1$*-dimensional space.*

any vector on the line, such as $\mathbf{p}$ or $\frac{1}{2}(\mathbf{p}+\mathbf{q})$ or the *perpendicular support vector* $\mathbf{d}$:

$$\mathbf{p} \wedge \mathbf{q} = \mathbf{p} \wedge (\mathbf{q}-\mathbf{p}) = \tfrac{1}{2}(\mathbf{p}+\mathbf{q}) \wedge (\mathbf{q}-\mathbf{p}) = \mathbf{d} \wedge (\mathbf{q}-\mathbf{p}) \tag{41}$$

where $\mathbf{d}$ is defined by $\mathbf{d} \wedge (\mathbf{q}-\mathbf{p}) = \mathbf{p} \wedge \mathbf{q}$ and $\mathbf{d} \cdot (\mathbf{q}-\mathbf{p}) = 0$. (These equations can be solved using the geometric product to give: $\mathbf{d} = (\mathbf{p} \wedge \mathbf{q})(\mathbf{q}-\mathbf{p})^{-1}$, a nice example of the use of division by vectors.)

So the outer product $p \wedge q$ can be used to represent a *directed line element* of the line pq. However, note that $p \wedge q$ is not a line *segment*: neither p nor q can be retrieved from $p \wedge q$. The 2-blade is just a line element of specified direction and length, somewhere along the line through p and q (in that order).

As a blade, we can use $p \wedge q$ to give an equation for the whole line: a point x is on the line through p and q if and only if $x \wedge (p \wedge q) = 0$. Let's verify that:

$$x \wedge p \wedge q = e_0 \wedge (\mathbf{p} \wedge \mathbf{q} - \mathbf{x} \wedge (\mathbf{q}-\mathbf{p})) + \mathbf{x} \wedge \mathbf{p} \wedge \mathbf{q} \tag{42}$$

This is zero if and only if two conditions hold: (1) $\mathbf{x} \wedge (\mathbf{q}-\mathbf{p}) = \mathbf{p} \wedge \mathbf{q} = \mathbf{p} \wedge (\mathbf{q}-\mathbf{p})$, so that $\mathbf{x} = \mathbf{p} + \lambda(\mathbf{q}-\mathbf{p})$ which is indeed the usual line equation; and (2) $\mathbf{x} \wedge \mathbf{p} \wedge \mathbf{q} = 0$ – but this holds when we have satisfied the first condition.

Geometrically, a point x lies on the line through p and q if the vector x in the homogeneous model lies in the plane spanned by p and q: eq.(42) is the statement that they span no volume. This is depicted in Figure 12b or c. You see that the geometry of homogeneous subspaces of 3-space is a faithful representation of the geometry of offset subspaces in 2-space. In the classical homogeneous model, one can only use this fact for the representation of *points*, since with vectors one can only span 1-dimensional subspaces representing 0-dimensional offset subspace. With geometric algebra, we can suddenly use this idea to describe any affine (i.e. offset) subspace. We simply continue this construction: an element of the oriented plane through the points p, q and r is represented by $p \wedge q \wedge r$, and so on for higher dimensional 'offset' subspaces – if the space has enough dimensions to accommodate them.

8.2.3 Equivalence of alternative characterizations

A special and rather satisfying property of this construction is its insensitivity to the kind of objects we use to construct the subspace. Of course the element of the line through p and q is determined by two points, or by a point and a direction. We would normally think of those as different constructions. However, in geometric algebra

$$p \wedge q = p \wedge (\mathbf{q} - \mathbf{p}) \tag{43}$$

(verify this!). So the two are exactly equal, they produce the same element by the same operation of 'taking the outer product'. Moreover, the intrinsic 'sliding' symmetry of the support vector (any of $\mathbf{p} + \lambda(\mathbf{q} - \mathbf{p})$ can be used) is also automatically absorbed in the representation $p \wedge q$ due to the 'sliding' symmetry of the outer product term $\mathbf{p} \wedge \mathbf{q}$ in it. For instance, we may rewrite it as $\mathbf{p} \wedge \mathbf{q} = \frac{1}{2}(\mathbf{p} + \mathbf{q}) \wedge (\mathbf{q} - \mathbf{p})$, showing that the midpoint $\frac{1}{2}(\mathbf{p} + \mathbf{q})$ is on the carrier line. We have in $p \wedge q$ just the right mixture of specificity and freedom to denote the desired geometric entity.

You may verify that in general, a k-dimensional subspace element B determined by the points at locations $\mathbf{p}_0, \cdots \mathbf{p}_k$ is represented in the homogeneous model by the $(k + 1)$-blade

$$B = p_0 \wedge \cdots \wedge p_k$$

and that this is equivalent, by the rules of computation for the outer product, to specifying it by a point and k directions

$$B = p_0 \wedge (\mathbf{p}_1 - \mathbf{p}_0) \wedge \cdots \wedge (\mathbf{p}_k - \mathbf{p}_0)$$

or any intermediate form specifying some positions and some directions. It is satisfying not to have to make different data-structures for those many ways of specifying this single geometrical object; the 'constructor' $\wedge$ takes care of it automatically. Testing of equivalence of various objects is therefore much simplified. The paper [12] goes on to use this to develop a complete 'simplicial calculus' for simplices specified in this manner, deriving advanced results in a highly compact algebraic and computational manner.

8.2.4 Intersection and incidence

The meet and join operations can be applied immediately to blades in the homogeneous model, and return blades representing the intersection and union of the corresponding Euclidean entities. Of course meet and join should be implemented as basic operations, but it pays to look in a little more detail how the various elements of the Euclidean results are packaged in a single homogeneous result, to get a feeling for the power of the representation. To do so we consider separate cases – but we emphasize that the meet and join themselves do not show such a breakup in cases explicitly: they are handled completely internally and automatically.

- *line and hyperplane*
  When intersecting a line with a hyperplane in general position (two lines in 2-space, a line and a plane in 3-space), the meet produces the unique intersection point, weighted by an 'intersection strength' denoting how perpendicular the intersection is, and hence how significant numerically.

  Let the line be $p \wedge \mathbf{u}$, and the hyperplane $q \wedge \mathbf{V}$, both in general position in m-dimensional space with pseudoscalar $\mathbf{I}$. Then their join is $\mathbf{I}_m$, and we get for their meet after some rewriting:

  $$(p \wedge \mathbf{u}) \cap (q \wedge \mathbf{V}) \quad = \quad e_0 \mathbf{u}^* \cdot \mathbf{V} + (\mathbf{p} \wedge \mathbf{u})^* \cdot \mathbf{V} - \mathbf{u}^* \cdot (\mathbf{V} \wedge \mathbf{q})$$

  (duality relative to $\mathbf{I}_m$), and this therefore represents the point at location

  $$\frac{(\mathbf{p} \wedge \mathbf{u})^* \cdot \mathbf{V} - \mathbf{u}^* \cdot (\mathbf{V} \wedge \mathbf{q})}{\mathbf{u}^* \cdot \mathbf{V}}$$

  So we obtain a clear geometrical entity as a result of such a meet, as long as $\mathbf{u}^* \cdot \mathbf{V} \neq 0$; which is the demand $\mathbf{u} \wedge \mathbf{V} \neq 0$ equivalent to the linear independence demand usually expressed as a determinant in the classical treatment. Note how the point is fully expressible in closed form, using only basic geometric operations.

- *parallel lines*
  Geometric algebra still gives consistent results when we compute the meet between subspaces that do not geometrically intersect in the classical sense.

  For instance, between *two parallel lines* $p \wedge \mathbf{u}$ and $q \wedge \mathbf{u}$, in a plane with 2-blade $\mathbf{I}$ determining their join and the corresponding duality, we get (after some rewriting):

  $$(p \wedge \mathbf{u}) \cap (q \wedge \mathbf{u}) = ((\mathbf{p} - \mathbf{q}) \wedge \mathbf{u})^* \mathbf{u},$$

  exhibiting the common directional part $\mathbf{u}$, weighted by a scalar magnitude proportional to the distance of the lines. This is still clearly interpretable, and more importantly, one can continue to compute with it since it is a regular element of the algebra. Its only unusual aspect is in its interpretation, not in its computational properties.

- *skew lines*
  Similarly, by a direct computation (see [4]), you may establish that *two skew lines* $p \wedge \mathbf{u}$ and $q \wedge \mathbf{v}$ in 3-dimensional space (which therefore have a join of $e_0 \wedge \mathbf{I}_3$ in the homogeneous model), have a meet of

  $$(p \wedge \mathbf{u}) \cap (q \wedge \mathbf{v}) = ((\mathbf{p} - \mathbf{q}) \wedge \mathbf{u} \wedge \mathbf{v})^*$$

  (with duality relative to $\mathbf{I}_3$). This is a scalar, proportional to the *perpendicular signed distance between the two lines* (weighted by the meet of their directions $\mathbf{u} \cap \mathbf{v} = (\mathbf{u} \wedge \mathbf{v})/\mathbf{I}_2$ in their common plane $\mathbf{I}_2$).

These examples suggest that the meet is not just an intersection operation: it is a general *incidence operation,* which computes the highest order geometric object in common between its arguments. That may be an actual offset subspace (as in the first example), or the scalar distance, possibly as a factor for common directional elements. All are legitimate outcomes in the full framework of geometric algebra, and we have to learn how to write algorithms using this new and stronger notion of incidence in its computation – it would prevent the splits into the different kinds of incidence which are required in the classical approach, and which are the potential source of so many errors.

8.3 The conformal model

A recently developed model of Euclidean space E^m is the *conformal model* $V^{n+1,1}$. This is a true *algebra of points*, or rather, *an algebra of spheres* (with points being spheres of zero radius). Again, points at locations $\mathbf{p}$ and $\mathbf{q}$ are represented by vectors p and q in the model, but now in a manner such that *the inner product represents their Euclidean distance*:

$$p \cdot q = -\tfrac{1}{2}(\mathbf{p} - \mathbf{q})^2 \tag{44}$$

In particular, $p \cdot p = 0$, so that points are represented by vectors which have – in their representative space – a zero norm! To do this and still have a complete geometric algebra requires *two* extra dimensions, so an m-dimensional Euclidean space is now represented using the geometric algebra of an $(m+2)$-dimensional space. Moreover, one of these extra dimensions is represented by a basis vector which squares to -1 (such spaces are known as Minkowski spaces).

A useful basis for this space is: an orthonormal basis for the Euclidean space embedded in it, and the vectors e_0 and e_∞ to represent the *point at the origin*, and the *point at infinity*, respectively. The two satisfy: $e_0 \cdot e_\infty = 1$, they are null vectors: $e_0 \cdot e_0 = 0$ and $e_\infty \cdot e_\infty = 0$, and they are orthogonal to the Euclidean subspace, so that $e_0 \cdot \mathbf{x} = 0$ and $e_\infty \cdot \mathbf{x} = 0$ for any $\mathbf{x} \in E^m$. The representation of a point p of Euclidean space in this conformal model is the vector:

$$p = e_0 + \mathbf{p} - \tfrac{1}{2}\mathbf{p}^2 e_\infty$$

(or a scalar multiple). You may verify that $p^2 = 0$, and that

$$p \cdot q = (e_0 + \mathbf{p} - \tfrac{1}{2}\mathbf{p}^2 e_\infty) \cdot (e_0 + \mathbf{q} - \tfrac{1}{2}\mathbf{q}^2 e_\infty) = -\tfrac{1}{2}\mathbf{q}^2 + \mathbf{p} \cdot \mathbf{q} - \tfrac{1}{2}\mathbf{p}^2 = -\tfrac{1}{2}(\mathbf{q} - \mathbf{p})^2$$

as desired.

Any point $\mathbf{x}$ on the hyperplane perpendicularly bisecting the line segment pq satisfies $(\mathbf{x} - \mathbf{p})^2 = (\mathbf{x} - \mathbf{q})^2$, and therefore:

$$x \cdot (q - p) = 0.$$

It follows that $q - p = (\mathbf{q} - \mathbf{p}) - \frac{1}{2}(\mathbf{q}^2 - \mathbf{p}^2)e_\infty$ *dually represents the midplane of p and q*, see eq.(20). In general, a hyperplane with orthogonal support vector $\mathbf{d}$ is (dually) represented by the vector

$$d = \mathbf{d}^{-1} - e_\infty$$

or any multiple of it, such as $\mathbf{n} - \delta e_\infty$ with $\mathbf{n}$ its normal vector and δ the support along $\mathbf{n}$ of the hyperplane. You may verify that the equation $x \cdot d = 0$ is indeed equivalent to the normal hyperplane equation $\mathbf{x} \cdot \mathbf{n} = \delta$.

8.3.1 Spheres are blades

The direct expression of the Euclidean distance by the inner product in eq.(44) implies that the equation

$$x \cdot c = -\tfrac{1}{2}\rho^2$$

is the equation of a sphere with radius ρ and center c. We rewrite this to

$$x \text{ on sphere with radius } \rho \text{ and center } c \quad \Longleftrightarrow \quad x \cdot (c + \tfrac{1}{2}\rho^2 e_\infty) = 0,$$

so this shows that *the vector $c + \frac{1}{2}\rho^2$ dually represents a sphere*. Where the homogeneous model can be used to code a hyperplane by a homogeneous normal vector, the conformal model (dually) represents a complete sphere by a single representative vector! In the conformal model, *(dual) spheres are basic elements of computation*. We get an algebra of spheres; a point is just a (dual) sphere of radius zero.

The direct (rather than dual) representation of a sphere is through the wedge product: *spheres are blades* in the conformal model. This is obvious since the dual of the vector $c + \frac{1}{2}\rho^2 e_\infty$ is an $(m-1)$-blade in the $(m+2)$-dimensional representation space. So, we have:

$$x \text{ on sphere through } p, q, r, s \quad \Longleftrightarrow \quad x \wedge (p \wedge q \wedge r \wedge s) = 0.$$

Moreover, the two representations are exactly dual in the conformal representation, so we can compute the center and radius of a sphere given by four points immediately through using:

$$p \wedge q \wedge r \wedge s = (c + \tfrac{1}{2}\rho^2 e_\infty)^*$$

It is very satisfying that these two totally different specifications of a sphere should be literally duals of each other, i.e. perpendicular to each other in the representative space of the conformal model. It is also a very pleasant surprise that the very complicated symmetries of four points determining the same sphere are simply reduced to the anti-symmetry of the outer product (as were the symmetries of the support vectors of hyperplanes in the homogeneous model). Spheres are not really new objects requiring totally new products – as long as you treat them in their own algebra, they behave just like subspaces.

We note that $p \wedge q$ is a 1-dimensional sphere, i.e. the computational representation of a point pair. In contrast to the homogeneous model, $p \wedge q$ now really has the semantics of a localized line *segment* rather than merely a line *element*.

8.3.2 Intersection of spheres

In the homogeneous model we saw that a factorization like eq.(43) gave literal equivalence of the same geometric object specified in different ways. Such simplifications also occur in the conformal model. Indeed, the dual equivalence of the sphere specifications just treated can be used in this way. Another example is the intersection of two spheres, which should produce a circle in a well-defined plane. Let us take a simple example, equal sized spheres of radius ρ at opposite sides $\pm\mathbf{c}$ of the origin. The dual of their intersection is computed as the outer product of their duals, which can then be rewritten in more convenient form:

$$(e_0 - \mathbf{c} - \tfrac{1}{2}(\mathbf{c}^2 - \rho^2)e_\infty) \wedge (e_0 + \mathbf{c} - \tfrac{1}{2}(\mathbf{c}^2 - \rho^2)e_\infty) = 2\mathbf{c} \wedge (e_0 + \tfrac{1}{2}(\mathbf{c}^2 - \rho^2)e_\infty)$$

The right hand side is immediately recognizable as the dual of the intersection of a hyperplane with normal $\mathbf{c}$ through the origin (its dual representation is $\mathbf{c}$) with a sphere at the origin of radius $\mathbf{c}^2 - \rho^2$. So these two alternative representations of the intersection circle are just two factorizations of the element of geometric algebra representing it (many other factorizations exist). Note how we can *compute directly with spheres and planes* rather than with equations asserting properties of points on it.

8.3.3 Unification of translations and rotations

The conformal model unites rotations and translations in a satisfying manner: both are representable as the exponent of a 2-blade. We have seen that the rotations require a 2-blade $\mathbf{I}\phi/2$ denoting a plane in the Euclidean space, and that a rotation can then be represented as

$$\text{rotation:} \quad \mathbf{x} \mapsto e^{-\mathbf{I}\phi/2}\mathbf{x}e^{\mathbf{I}\phi/2}.$$

A translation turns out to be representable as the exponent of a 2-blade $e_\infty \wedge \mathbf{p}/2$ containing the point at infinity and the translation vector $\mathbf{p}$. Because e_∞ squares to zero and commutes with $\mathbf{p}$, we obtain

$$e^{e_\infty \mathbf{p}/2} = 1 + e_\infty \mathbf{p}/2.$$

You can now use this to verify that the translation of the point at the origin (represented by e_0) indeed gives the point at $\mathbf{p}$:

$$p = e^{-e_\infty \mathbf{p}/2} e_0 e^{e_\infty \mathbf{p}/2} = e_0 + \mathbf{p} - \tfrac{1}{2}\mathbf{p}^2 e_\infty.$$

Having rotations and translations in the same form permits a concise treatment of rigid body motions, presenting new unifying insights in traditional representations such as screws [9]. This may well transfer them from theoretical mechanics to practical computational geometry, as the next refinement after quaternions.

9 Conclusion

This introduction of geometric algebra intends to alert you to the existence of a limited set of products that appears to generate all geometric constructions in one consistent framework. Using this framework can simplify the set of data structures representing objects since it inherently encodes all relationships and symmetries of the geometrical primitives in those operators (an example was eq.(41)). Also, it could serve as a straight-jacket for the specification of geometric algorithms, preventing the unbridled invention of new operations and objects without clear and clean geometrical meaning, or well-defined relationships to other objects in the application. If our hopes are correct, this straight-jacket would actually *not* be a limitation on what one can construct; rather it contains precisely the right set of operations to provide a precise language for arbitrary constructions. The basic operations even have the power to model the geometry of spheres and their interactions; thus the same syntax admits of varied semantics.

That such a system exists is a happy surprise to all learning about it. Whether it is also the way we should structure our programming is at the moment an open question. Use of the conformal model would require representing the computations on the Euclidean geometry of a 3-dimensional space on a basis of $2^{3+2} = 32$ elements, rather than just 3 basis vectors (plus 1 scalar basis). It seems a hard sell. But you often have to construct objects representing higher order relationships between points (such as lines, planes and spheres) anyway, even if you do not encode them on such a 'basis'. Also, our investigations show that perhaps all one needs to do all of geometry are blades and operators composed of products of vectors; the product combinations of this limited subset can be optimized in time and space requirements, with very little overhead for their membership of the full geometric algebra. That automatic membership would enable us to compute directly with lines, planes, circles and spheres and their intersections without needing to worry about special or degenerate cases, which should eliminate major headaches and bugs. We also find the coordinate-free specification of the operations between objects very attractive; relegating the use of coordinates purely to the input and output of geometric objects banishes them from the body of the programs and frees the specification of algorithms from details of the data structures used to implement them. Such properties makes geometric programs so much more easy to verify, and – once we have learned to express ourselves fluently in this new language – to construct.

We are currently investigating these possibilities, doing our best to make the geometric algebra approach a reasonable alternative. The main delay now is that the algebra dictates a new way of thinking about geometry which requires one to revisit many old constructions. This takes time, but is worthwhile since it appears to simplify the whole structure of geometric programming. At the very least, we would hope geometric algebra to be a useful meta-language in which to specify geometric programs; but the proven efficiency of quaternions, which are such a natural part of geometric algebra, suggests that we might even want to do our low-level computations in this new computational framework.

10 Further reading

There is a growing body of literature on geometric algebra. Unfortunately much of the more readable writing is not very accessible, being found in books rather than journals. Little has been written with computer science in mind, since the initial applications have been to physics. No practical implementations in the form of libraries with algorithms yet exist (though there are packages for Maple [1] and Matlab [5] which can be used as a study-aid or for algorithm design). We would recommend the following as natural follow-ups on this paper:

- GABLE: a Matlab package for geometric algebra, accompanied by a tutorial

[5].

- The introductory chapters of 'New Foundations of Classical Mechanics' [8].
- An introductory course intended for physicists [2].
- An application to a basic but involved geometry problem in computer vision, with a brief introduction into geometric algebra [11].
- A paper showing how linear algebra becomes enriched by viewing it as a part of geometric algebra: [10].

If you read them in approximately this order, you should be alright. We are working on texts more specifically suited for a computer graphics audience; these will probably first appear as SIGGRAPH courses.

References

[1] A. Lasenby, M. Ashdown et al., *GA package for Maple V*, 1999, available at `http://www.mrao.cam.ac.uk/~clifford/software/GA/`

[2] C. Doran and A. Lasenby, *Physical Applications of Geometric Algebra*, 2001, available at `http://www.mrao.cam.ac.uk/~`clifford/ptIIIcourse/

[3] C. Doran, A. Lasenby, S. Gull, *Chapter 6: Linear Algebra*, in: Clifford (Geometric) Algebras with applications in physics, mathematics and engineering, W.E. Baylis (ed.), Birkhäuser, Boston, 1996.

[4] L. Dorst, *Honing geometric algebra for its use in the computer sciences*, in: Geometric Computing with Clifford Algebra, G. Sommer, editor;, Springer ISBN 3-540-41198-4, expected 2000, Preprint available at `http://www.wins.uva.nl/~leo/clifford/`

[5] L.Dorst, S.Mann, T.A.Bouma, GABLE: a Geometric AlgeBra Learning Environment, `www.science.uva.nl/~leo/clifford/gable.html`

[6] R. Goldman, *Illicit Expressions in Vector Algebra*, ACM Transactions of Graphics, vol.4, no.3, 1985, pp. 223—243.

[7] R. Goldman, *The Ambient Spaces of Computer Graphics and Geometric Modeling*, IEEE Computer Graphics and Applications, vo.20, pp. 76–84, 2000.

[8] D. Hestenes, *New foundations for classical mechanics*, 2nd edition, D. Reidel, Dordrecht, 2000.

[9] D. Hestenes, *Old wine in new bottles*, in: Geometric Algebra: A Geometric Approach to Computer Vision, Quantum and Neural Computing, Robotics and Engineering, Bayro-Corrochano, Sobczyk, eds, Birkhäuser, to be published 2000, Chapter 24, pp. 498-520.

[10] D. Hestenes, *The design of linear algebra and geometry*, Acta Applicandae Mathematicae 23: 65-93, 1991.

[11] J. Lasenby, W. J. Fitzgerald, C. J. L. Doran and A. N. Lasenby. New Geometric Methods for Computer Vision Int. J. Comp. Vision 36(3), p. 191-213 (1998).

[12] G. Sobczyk, *Simplicial Calculus with Geometric Algebra*, in: Clifford Algebras and their Applications in Mathematical Physics, A. Micali et al. eds., Kluwer 1993.

[13] J. Stolfi, *Oriented projective geometry*, Academic Press, 1991.

Applications of Geometric Algebra

Joan Lasenby* & Chris Doran**

*Dept of Engineering, **Dept of Physics

University of Cambridge

http://www-sigproc.eng.cam.ac.uk/vision

http://www.mrao.cam.ac.uk/clifford

SIGGRAPH 2001 CREATE INTERACTION

Overview

Geometric Algebra

- Reciprocal frames
- Rotations: rotors, quaternions, Euler angles etc.
- Multivector differentiation
- Rotor interpolation

Applications relevant to computer graphics

- Computer Vision: camera calibration & 3d reconstruction
- Tracking and 3d skeleton-fitting using models
- Inverse kinematics and dynamics
- A framework for ray tracing

Reciprocal Frames

If $\{e_i\}$ forms a basis for an n-D space then $\{e^j\}$ is said to be the **reciprocal** basis and satisfies

$$e^i \cdot e_j = \delta^i_j$$

Basis not necessarily orthonormal.

Reciprocal basis is easily formed via following

$$e^j = (-1)^{j-1} e_1 \wedge \ldots \wedge \breve{e}_j \wedge \ldots \wedge e_n I_e^{-1}, \qquad I_e = e_1 \wedge e_2 \wedge \ldots \wedge e_n$$

Now have very useful property that:

$$a = a^i e_i = a_i e^i$$
$$(a^i = a \cdot e^i, \ \ a_i = a \cdot e_i)$$

Will use reciprocal frames later for intersecting rays with planar facets.

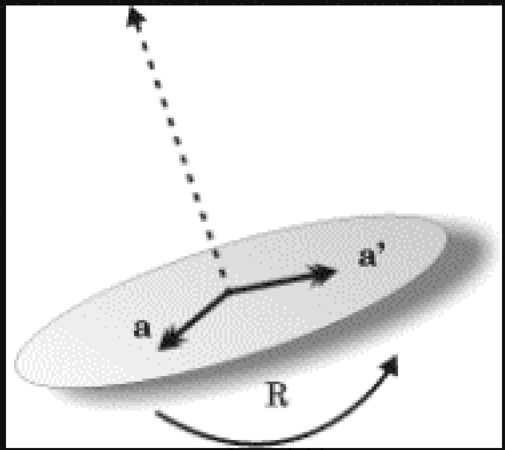
In what follows, † will refer to reversion: $A = abc \qquad A^\dagger = cba$

Using Rotors to rotate

- For our applications rotations are the most important aspect of GA – recall (work in 3D here)

$$a' = RaR^{\dagger} \qquad R = \exp(-B) = \cos\frac{\theta}{2} - I\hat{n}\sin\frac{\theta}{2}$$

$$B = \frac{\theta}{2}I\hat{n} \qquad RR^{\dagger} = 1$$



B (bivector) = plane of rotation

- Compare with quaternions:

$$q = a_o + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$$

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \quad \longleftrightarrow \quad \mathbf{i} = Ie_1, \quad \mathbf{j} = -Ie_2, \quad \mathbf{k} = Ie_3$$

The i,j,k of quaternions are therefore just the elementary bivectors

Rotors and Euler Angles

Euler angles: one commonly used convention follows

$$R_\phi = e^{-(I/2)\phi e_3}, \quad R_\theta = e^{-(I/2)\theta e_1'}, \quad R_\psi = e^{-(I/2)\psi\, e_3''}$$

$$e_1' = R_\phi e_1 R_\phi^\dagger \qquad e_3'' = R_\theta R_\phi e_1 R_\phi^\dagger R_\theta^\dagger$$

so

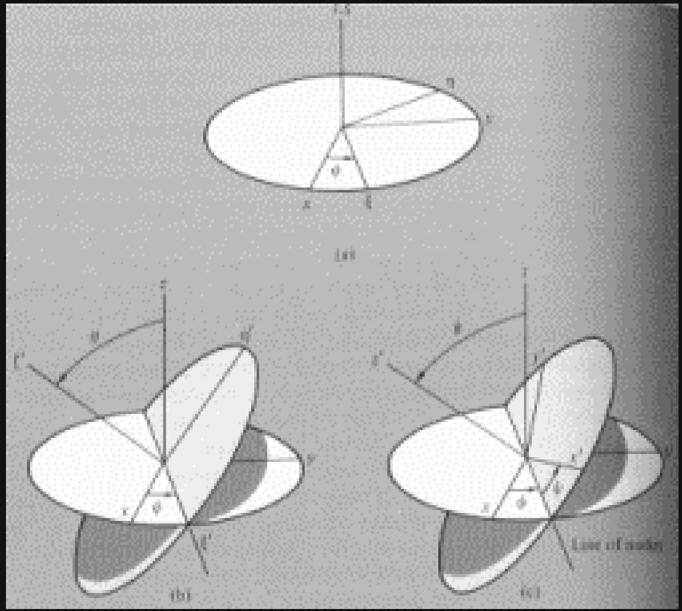
Then we are told to combine the rotation matrices as

$$x \longrightarrow R_\psi R_\theta R_\phi x R_\phi^\dagger R_\theta^\dagger R_\psi^\dagger$$

$$\mathbf{x} \longrightarrow \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi \mathbf{x} \qquad \mathbf{R}_\psi \equiv e^{-(I/2)\psi e_3} \quad etc$$

i.e. we combine the rotations about the **unrotated axes** in the *reverse* order.

This can be justified in a number of ways but is particularly simple using GA:



Goldstein(2nd Ed) P.146

Rotors and Euler Angles contd..

We can see this by taking

$$
\begin{aligned}
R_\theta R_\phi &= (\cos\frac{\theta}{2} - IR_\phi e_1 R_\phi^\dagger \sin\frac{\theta}{2})R_\phi \\
&= R_\phi(\cos\frac{\theta}{2} - Ie_1 \sin\frac{\theta}{2})R_\phi^\dagger R_\phi \\
&= R_\phi R_\theta'
\end{aligned}
$$

where R_θ' is a rotation of θ about the unrotated e_1 axis

The elements of the rotation matrix, R_{ij}, corresponding to the rotor R are given by the following very simple expressions

$$
\mathrm{R}_{ij} = e_i \cdot R e_j R^\dagger
$$

Converting between representations

Once we have our rotor description of a rotation, converting to any other representation is very easy:

Rotation matrix: $\mathsf{R}_{ij} = e_i \cdot R e_j R^\dagger$

Axis-Angle $\theta = 2\cos^{-1}\langle R\rangle_0, \quad \hat{n} = I\langle R\rangle_2/\sin(\theta/2)$

Quaternions $Q = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$

$a_0 = \langle R\rangle_0 \quad a_1 = (I\langle R\rangle_2)\cdot e_1 \quad a_2 = -(I\langle R\rangle_2)\cdot e_2 \quad a_3 = (I\langle R\rangle_2)\cdot e_3$

Euler angles $R = R'_\theta R'_\phi R'_\psi$ as before

$\phi = \cos^{-1}(\mathsf{R}_{33}) \quad \theta = \sin^{-1}(-\mathsf{R}_{13}/\sin\phi) \quad \psi = \sin^{-1}(\mathsf{R}_{31}/\sin\phi)$

Multivector Differentiation

- In GA we can differentiate wrt any element of the algebra
- For A, X multivectors, τ a scalar, F *a* general (multivector-valued function), derivative in 'direction' of A is

$$A * \partial_X F(X) \equiv \lim_{\tau \to 0} \frac{F(X + \tau A) - F(X)}{\tau}$$

$*$ => scalar part of product

- Extend this to **multivector derivative** by using a basis and reciprocal basis for the entire algebra $\{e_J\}$, $\{e^J\}$

$$\partial_X \equiv \sum_J \mathbf{e}^J (\mathbf{e}_J * \partial_X)$$

J summed over grades of X

Some examples.....

- Consider differentiating a bivector B wrt itself

$$\begin{aligned}\partial_B B &= \sum_{J\in B} \mathbf{e}^J(\mathbf{e}_J * \partial_B)B \\ &= \sum_{J\in B} \mathbf{e}^J \lim_{\tau\to 0} \frac{(B+\tau\mathbf{e}_J)-B}{\tau} = \sum_J \mathbf{e}^J\mathbf{e}_J = 3\end{aligned}$$

- Similarly it is not hard to show that

$$\partial_B B^2 = 2B, \quad \partial_B B^3 = 5B^2, \ldots$$

Will use this later….

- Replace A by $\boldsymbol{a}$ and $\boldsymbol{e}_J$ by $\boldsymbol{e}_j$ to get the **vector derivative**

$$\partial_a = \mathbf{e}^i \frac{\partial}{\partial a^i} \qquad \text{where} \qquad \boldsymbol{a} = a^i \mathbf{e}_i$$

Rotor Calculus

- Rotor differentiation will be key to many of our applications. Trick is to replace R by a general even element (spinor) ψ and differentiate wrt ψ – avoids messy differential geometry.

$$Ra\tilde{R} = \psi a \psi^{-1}$$

- Consider the scalar function of R:

$$S = (Ra\tilde{R}) \cdot b = \langle Ra\tilde{R} b \rangle \equiv \langle \psi a \psi^{-1} b \rangle$$

- It is not hard to show

$$\partial_\psi S = \partial_\psi \langle \psi a \psi^{-1} b \rangle = a\psi^{-1} b - \psi^{-1} b \psi a \psi^{-1}$$

- Leading to

$$\partial_R S = 2\tilde{R}\{(Ra\tilde{R}) \wedge b\}$$

Intuitively reasonable and important in least squares problems

Interpolation of Rotations

- Given two rotations, R_0 and R_1 say, can we define any point of the rotor path in a sensible way? We know that R takes us from R_0 to R_1 :

$$\boxed{R_1 = RR_0 \Rightarrow R = R_1\tilde{R}_0 \equiv \exp(B)}$$

- Yes, simply interpolate the bivector B – resulting path is rotationally invariant (transform endpoints and path transforms in same way).

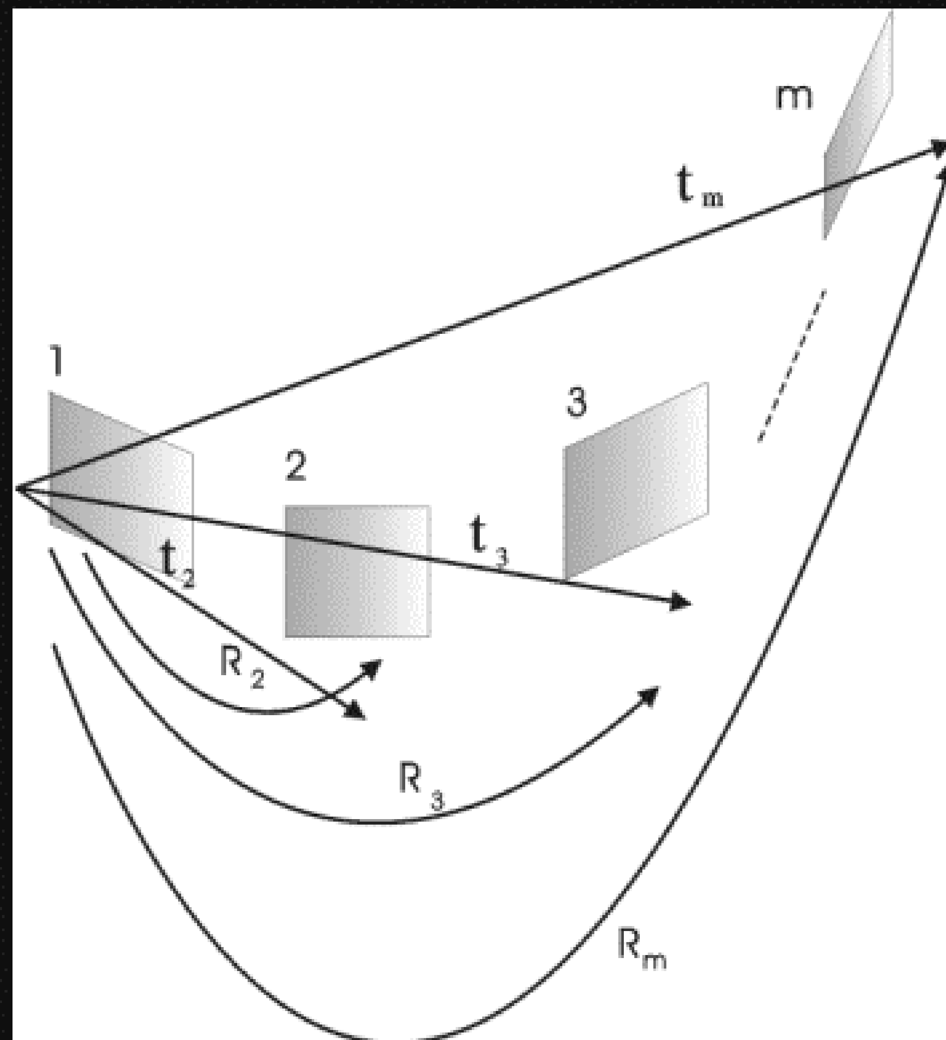
$$\boxed{\begin{array}{l} R(\lambda) = \exp(\lambda B)R_0 \\ R(0) = R_0, \quad R(1) = R_1 \end{array}}$$

← known about by those who use quaternions/rotational pseudovectors

2. Computer Vision: camera calibration

External calibration:

- Many applications require systems of multiple cameras – motion capture, visual surveillance etc.
- Need an easy way of finding camera positions.
- Let R_i, t_i be rotor and translation taking frame at camera 1 to frame at camera i.

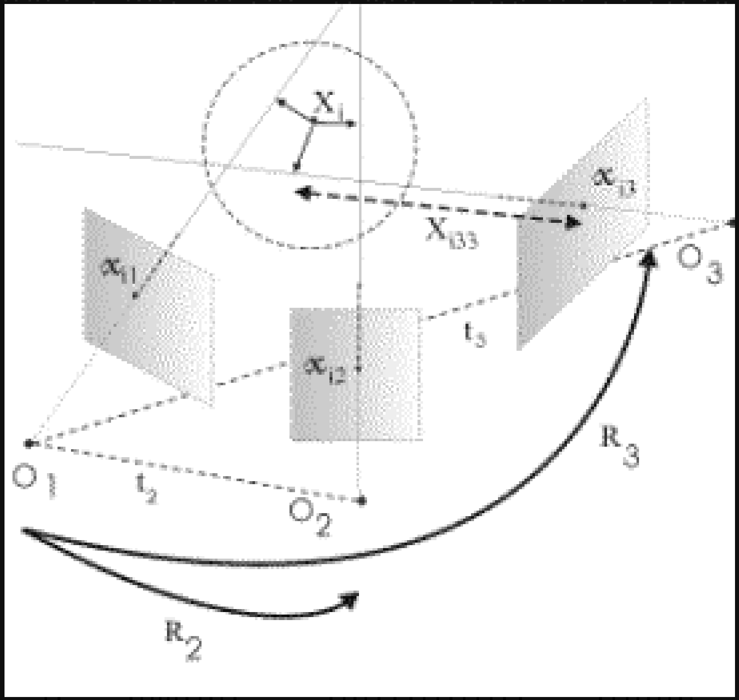


Maximum Likelihood Method

- Have n world points, X_i ($=X_{i1}$), in m cameras (X_{ij}): observe image points, x_{ij}, in camera j. O_{ij}=1/0 if point i visible/not visible in camera j.

- Since X_{ij} and X_i are related by

$$X_{ij} = \tilde{R}_j (X_i - t_j) R_j$$



- the following geometric cost function is suggested

$$S = \sum_{i=1}^{n} \sum_{j=1}^{m} \left[X_{ij3} x_{ij} - \tilde{R}_j (X_i - t_j) R_j \right]^2 O_{ij}$$

Optimisation Scheme

- Form derivatives and equate to zero:

$$\partial_{R_j} S = 0, \quad \partial_{t_j} S = 0,$$
$$\partial_{X_{ij3}} S = 0, \quad \partial_{X_i} S = 0,$$

- Involves **rotor differentiation** and **vector differentiation** – all carried out in a coordinate-free manner.
- Subsititute the maximum likelihood estimators for X_{ij3}s and X_is
- Solve resulting equations for Rs and ts iteratively.
- Converges in around 10 iterations if data is not poor and involves no search. Same algorithm for any m. Now used routinely in our motion capture system.

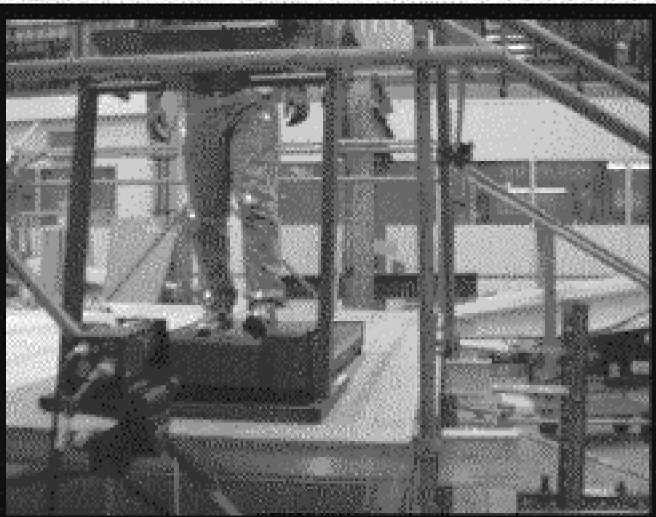
Motion Capture...

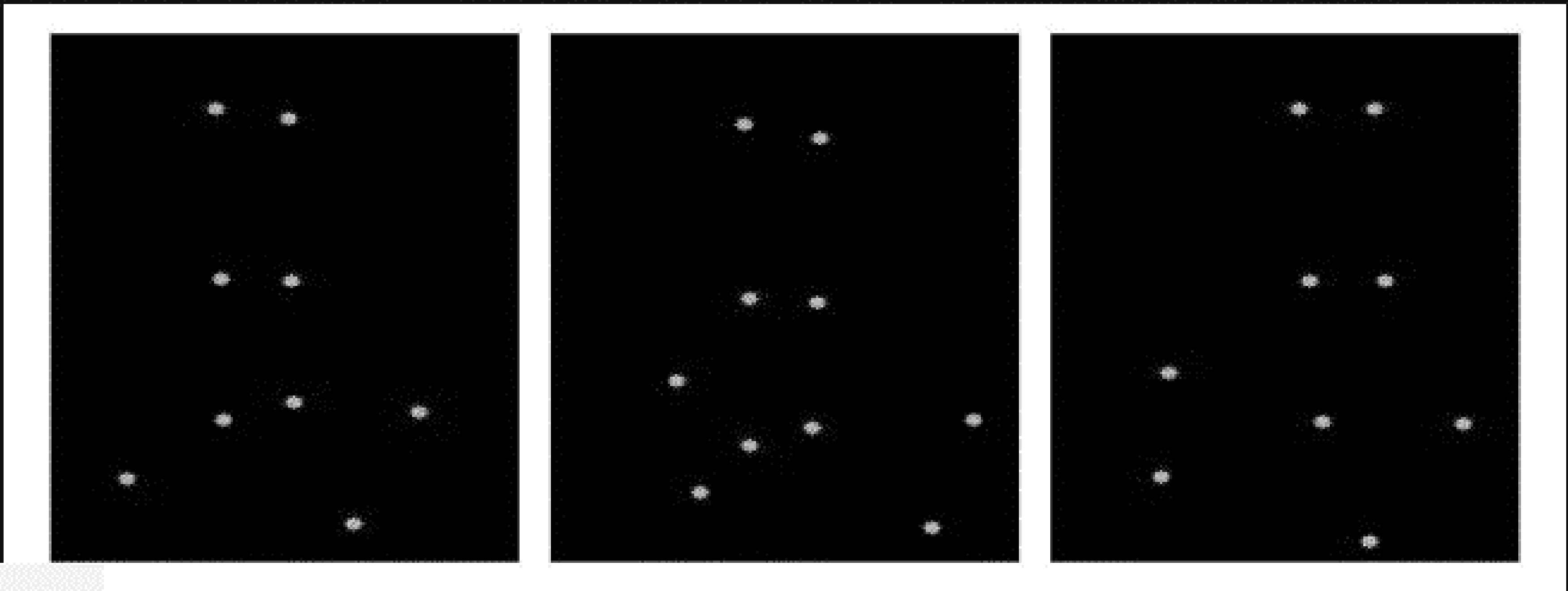
Once the cameras are calibrated we can use this to reconstruct 3D coordinates of tracked points: nice least squares/GA reconstruction method drops out of ML algorithm.

But the tracking itself can sometimes present the biggest problem……

Typical Output

Output from 3 cameras of a multicamera motion capture system: taken from a subject on mock-up of the Millennium Bridge (London)

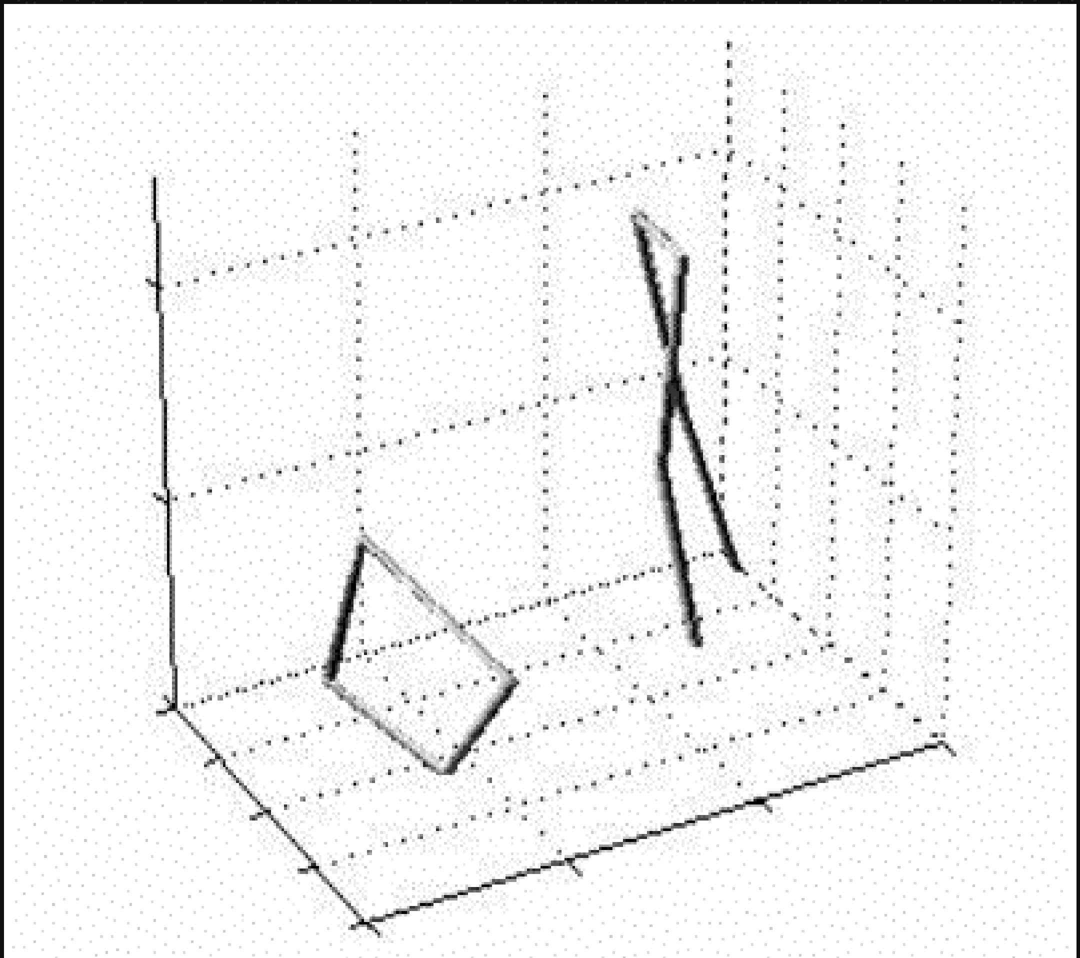




Video

Motion Reconstruction

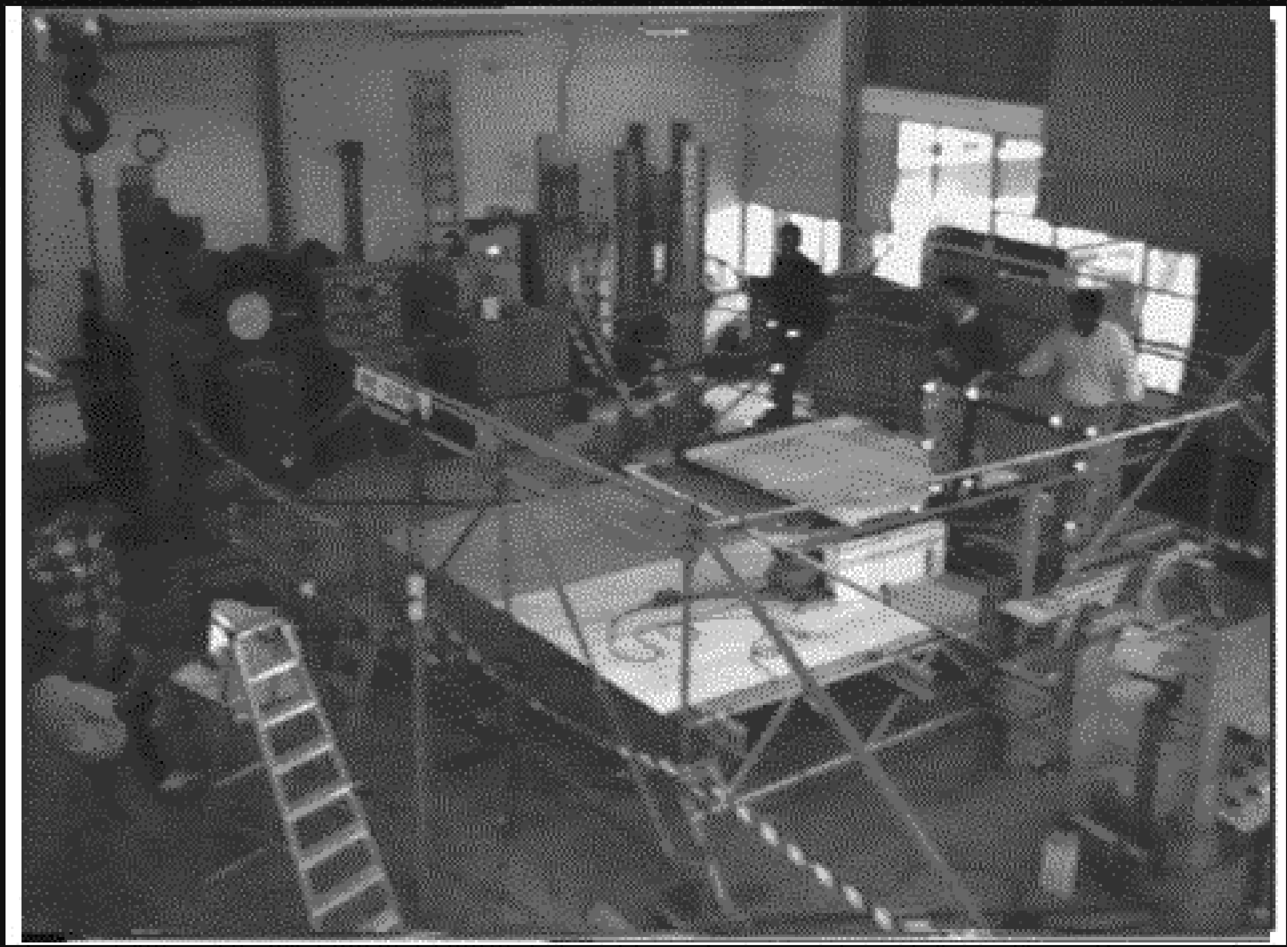
The single person on the moving bridge is simple to capture and reconstruct:



Video

Several independently moving objects

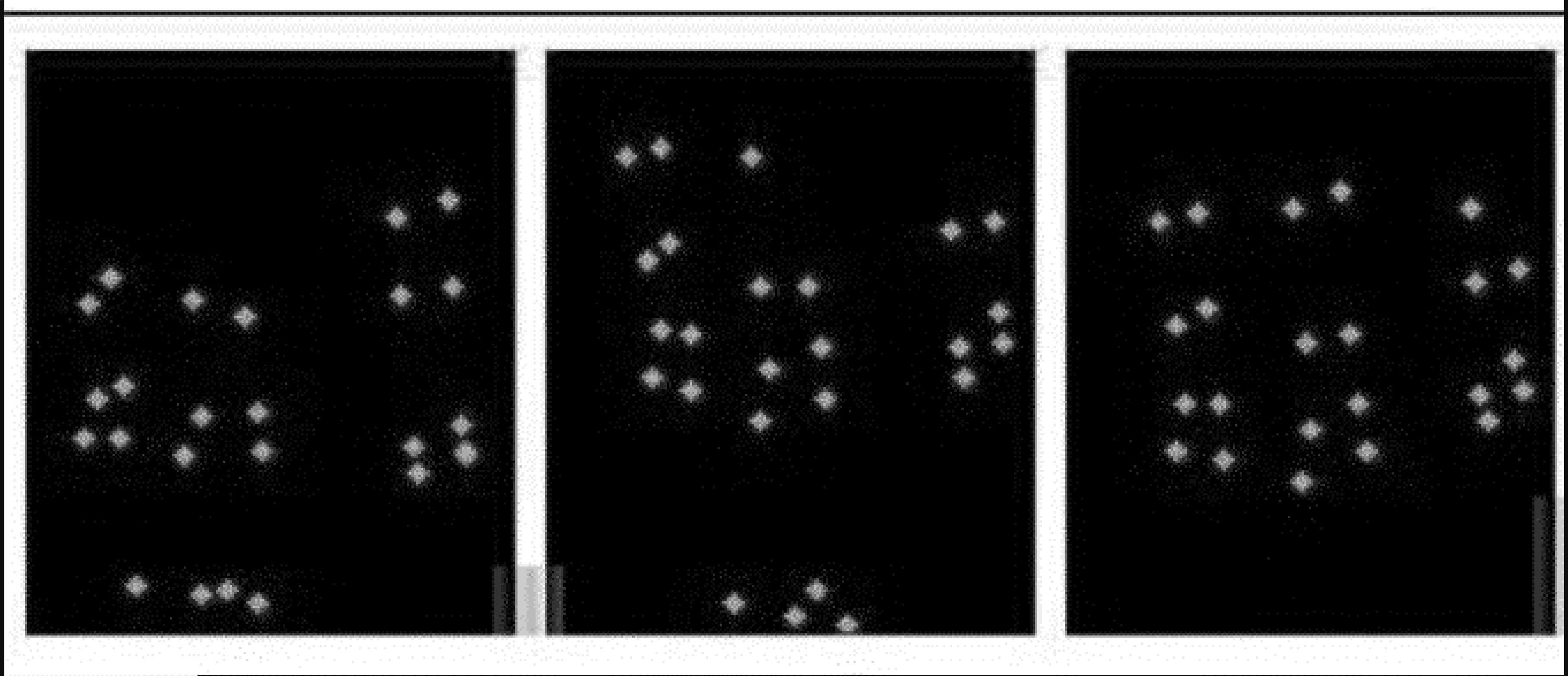
But things get more difficult to track when we have few cameras (25 or 50Hz) and several independently moving objects



Video

The camera output……

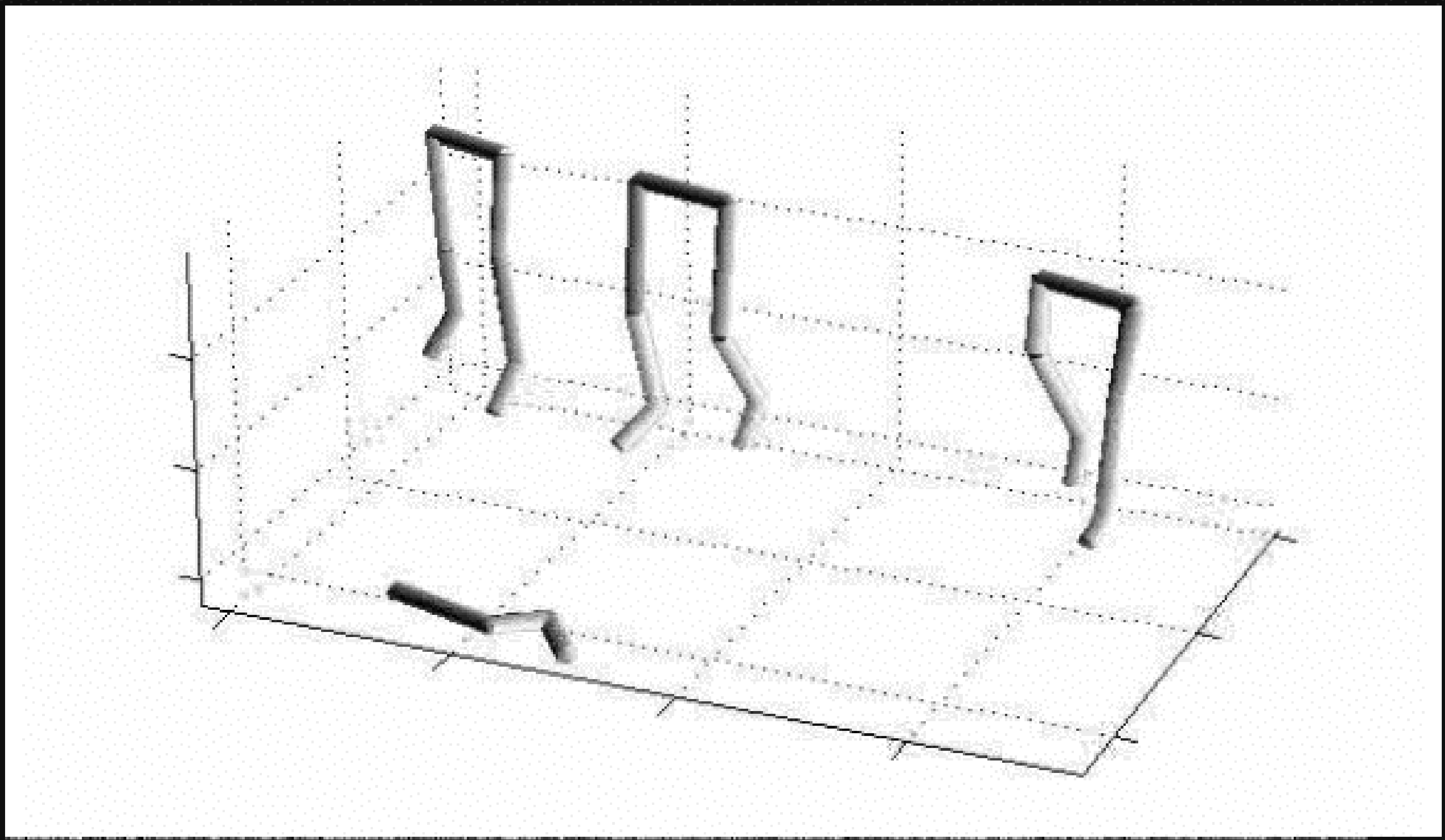
Output from 3 cameras taken from three subjects on mock-up of the Millennium Bridge – now we have more occlusions and whole dataset is much harder to track…..



Video

Motion Reconstruction

Here, the tracking and reconstruction proceeded without human intervention when models of the objects were used – the models are written in GA and the tracker tracks bivector components….



Video

Tracking and Articulated Models

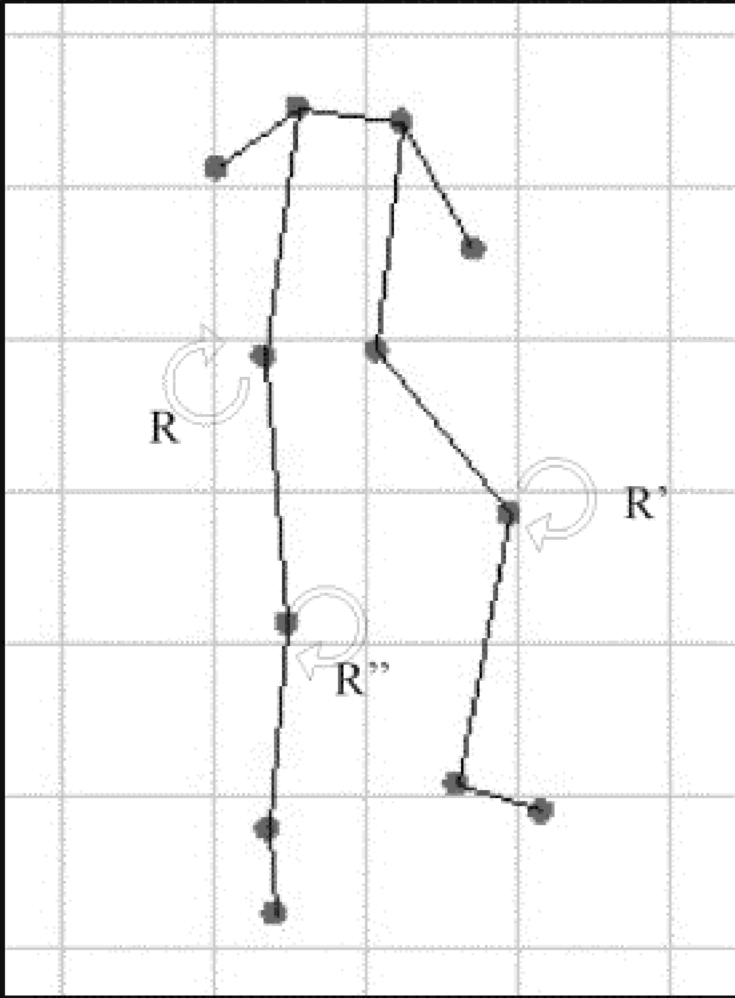
Need a good articulated model of our subject (plus some occlusion modelling) to make tracking simpler– we build these models using GA.

Forward kinematics:

Choosing a model and describing the observations in terms of the model.

Inverse kinematics:

Deducing the model parameters from the observations.



GA can simplify both the above

Articulated Models contd.....

A simple example – 2 jointed rods to model a leg. Can write:

$$x_b = x_a + R_1(d_1 e_1)R_1^\dagger \qquad R_1 = e^{-(I/2)\theta_1 n_1}$$

$$x_c = x_b + R_2 R_1(d_2 e_1)R_1^\dagger R_2^\dagger \qquad R_2 = e^{-(I/2)\theta_1 (R_1 e_2 R_1)}$$

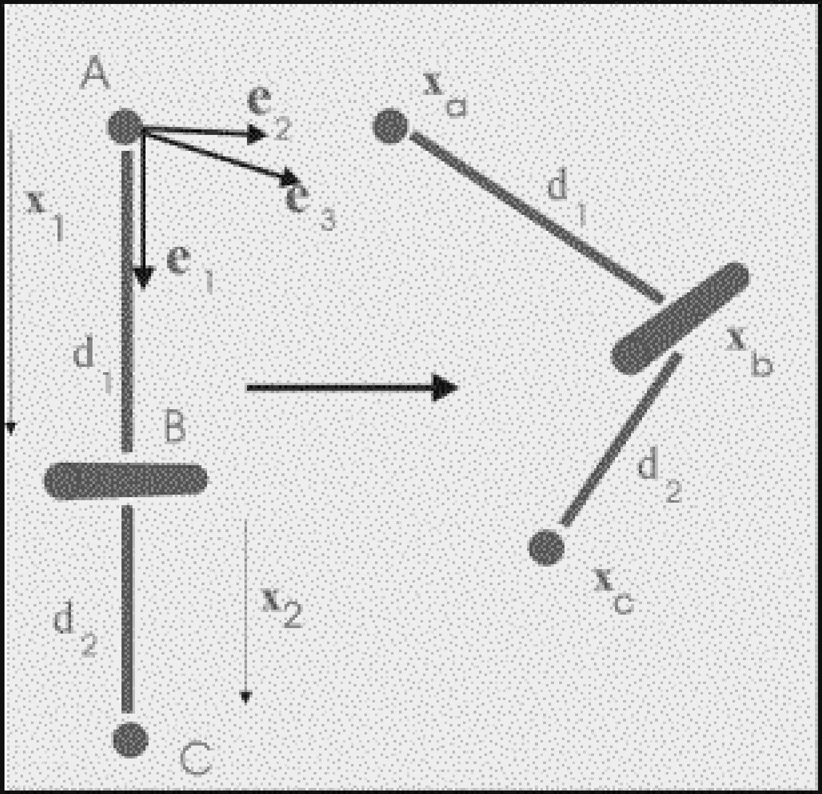
Transforming to non-rotated axes then gives:

$$x_b = x_a + R_1'(d_1 e_1)R_1'^\dagger$$

$$R_1' = e^{-(I/2)\theta_1 n_1}$$

$$x_c = x_b + R_1' R_2'(d_2 e_1)R_2'^\dagger R_1'^\dagger$$

$$R_2' = e^{-(I/2)\theta_1 e_2}$$

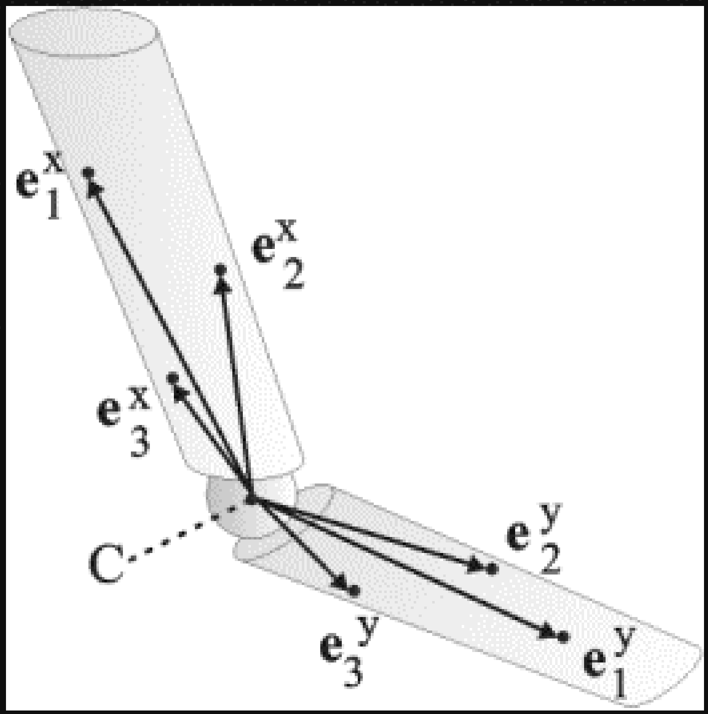


Process very straightforward.

Modelling contd..

Given these equations we can track markers placed on the joints using an EKF (extended Kalman filter) or particle filter with the rotation parameters (later we compare using bivectors with Euler angles) as our state variables. We can also use the model to predict occlusions.

But in reality we cannot place markers at the joints – and, in order to get *real* information on the rotation of the joint (possible rotation about axis of bone) we need off-axis markers.

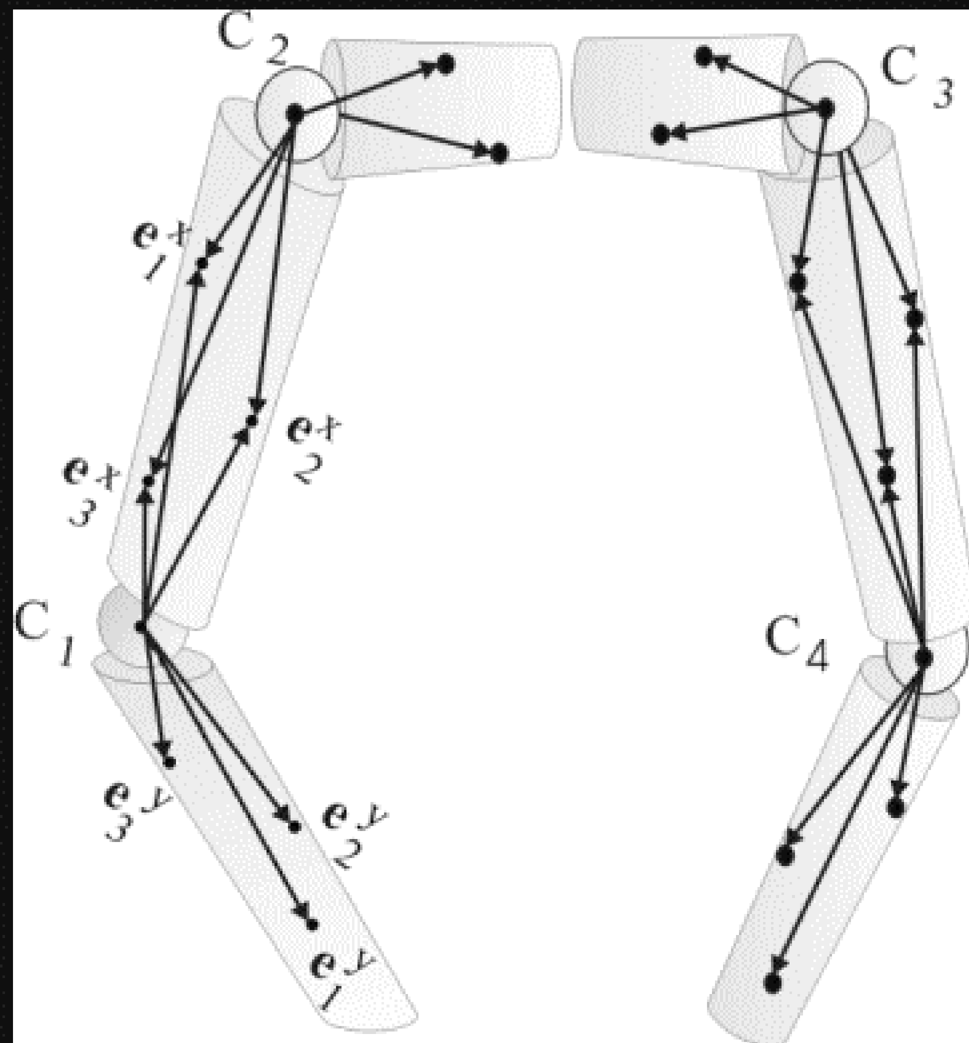


Need to estimate the $\{e_i^x\}$ *and* $\{e_j^y\}$

Skeleton Fitting

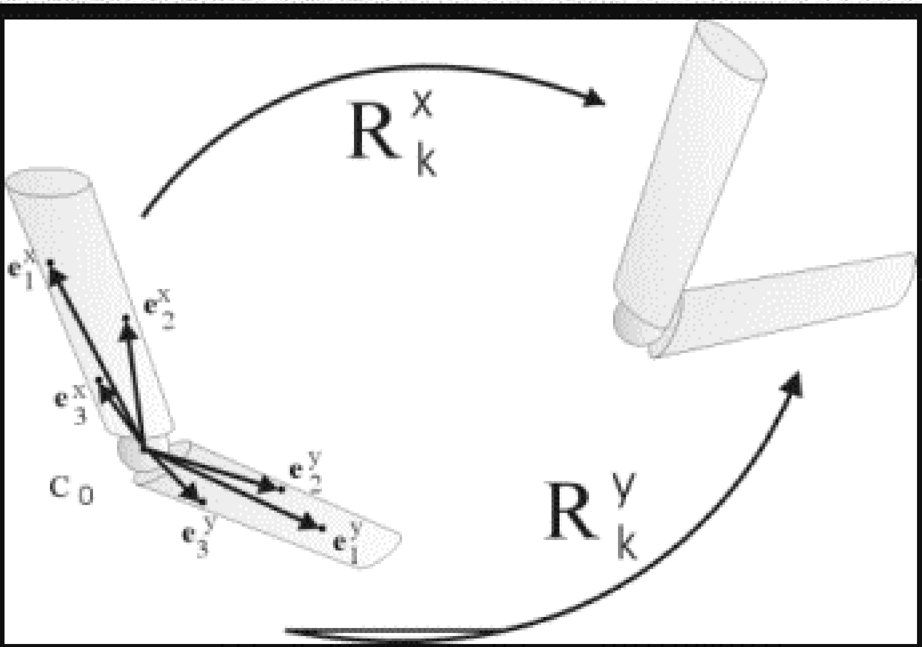
Therefore, we wish to determine, as accurately as possible, the dimensions of the skeleton of our subject and the positions of the markers relative to the joints

Since this model will be used for tracking, we need to extract it from some calibration data which is simple to track.



Skeleton Fitting cont...

First consider just 2 jointed limbs X and Y on which markers are placed at $\{e^x_i\}$ and $\{e^y_j\}$. Let x^k_i and y^k_j be the 3d coordinates (which we assume we can estimate) at time k. Then we know that



$$x^k_i = c^k_0 + R^x_k e^x_i R^{x\dagger}_k \quad and \quad y^k_j = c^k_0 + R^y_k e^y_j R^{y\dagger}_k$$

so that: $$b^k_{ij} = x^k_i - y^k_j = R^x_k e^x_i R^{x\dagger}_k - R^y_k e^y_j R^{y\dagger}_k$$

therefore find $\{e^x_i\}$ and $\{e^y_j\}$ that minimise

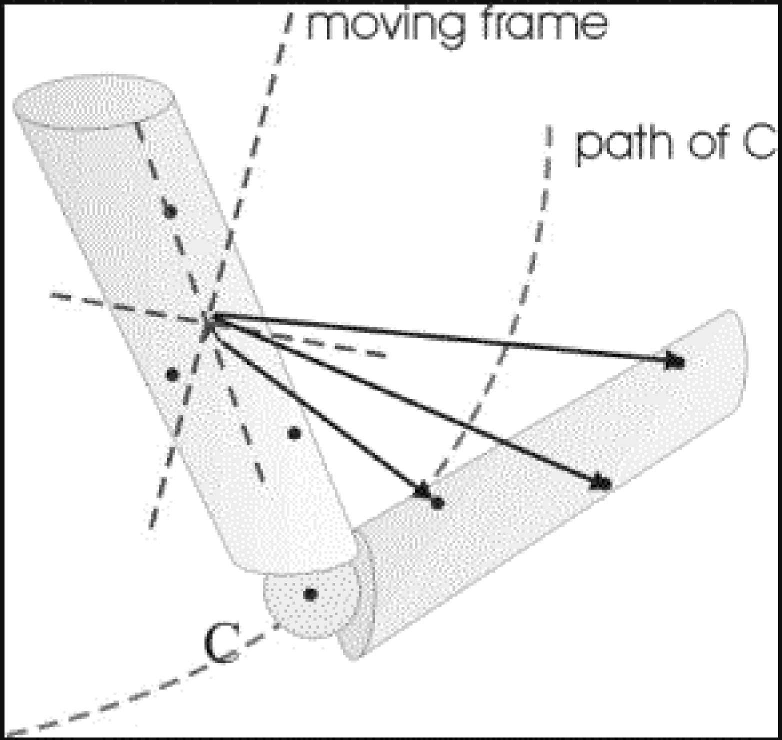
$$S = \sum_k \sum_i \sum_j [(x^k_i - y^k_j) - (R^x_k e^x_i R^{x\dagger}_k - R^y_k e^y_j R^{y\dagger}_k)]^2$$

Evidence that the errors are approx. Gaussian, so least squares is appropriate. We solve this by the following approach

- Substitute in for the MLEs for rotors R^x_k and R^y_k – found by using the estimation method described earlier for camera calibration.
- Form the derivative equations $$\partial_{e_i^x} S = 0, \quad \partial_{e_j^y} S = 0,$$
- Gives closed form solutions of the form
$$M^x a_x = b_x \qquad M^y a_y = b_y$$
$$a_x = [e^x_{11}, e^x_{12}, \ldots\ldots], \quad a_y = [e^y_{11}, e^y_{12}, \ldots\ldots]$$
- Giving very robust solutions – significantly better than many conventional methods:
- Once we have the $\{e\}$s we can estimater CoR.

Conventional CG/BM method

Currently in the computer graphics and biomechanical literature, methods which do the estimations for a ***fixed CoR*** are used, in conjunction with measurements relative to frame moving with one of the limbs.



Leads to significantly poorer estimates unless the points are manually weighted (immediately leading to rather ad-hoc algorithms).

Tracking using bivectors

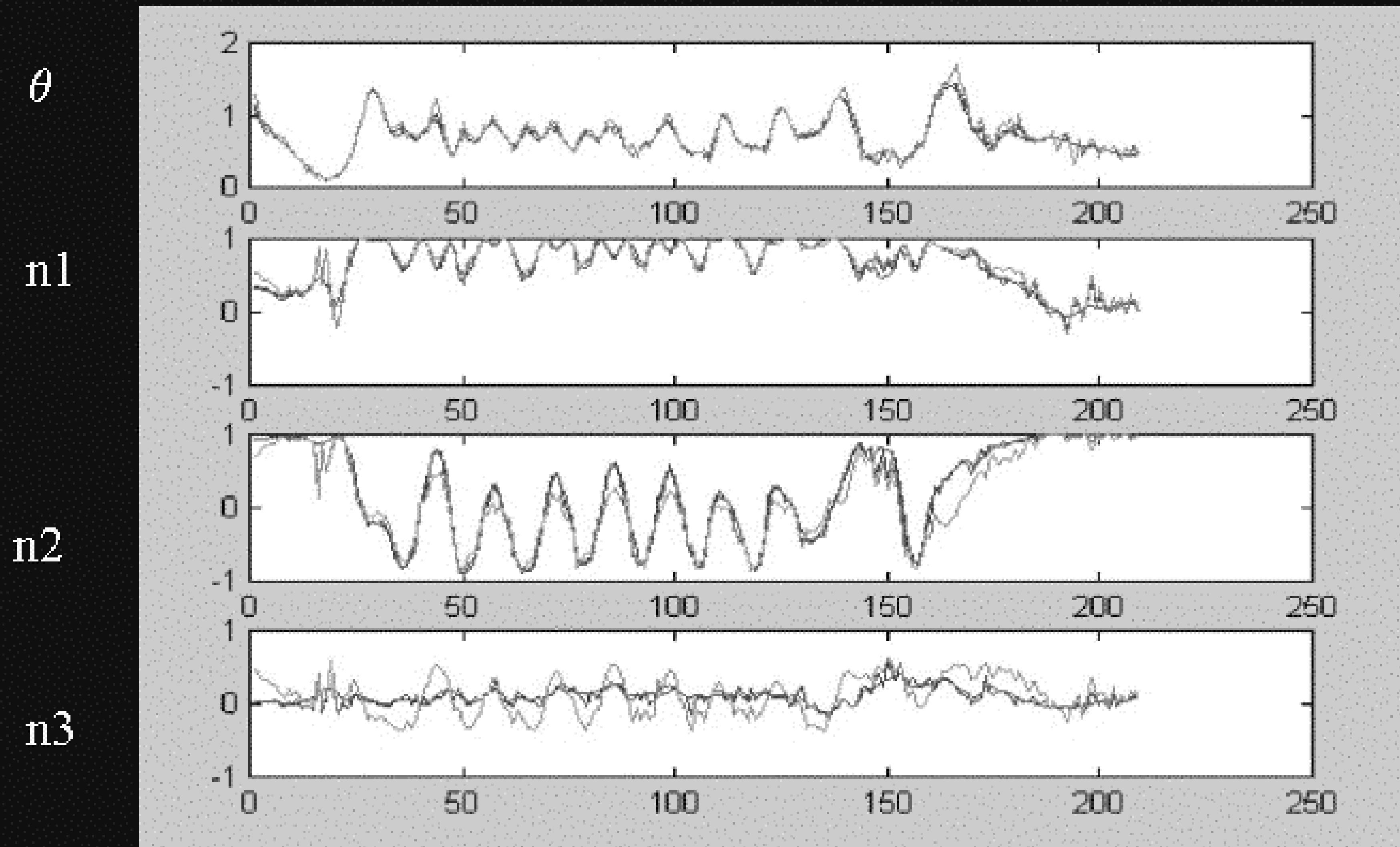
Now, from what has gone before, we can assume that we have a fully calibrated camera system and a fully articulated model of our subject's skeleton and markers – we now track our subject:

On the next slide we see the results of comparing two tracking scenarios – one using Euler angles, and the other using bivector components (need derivatives wrt bivectors).

Generate data so that there are large rotations involved (for small rotations there is little difference between methods). Add noise and track the data a) using Euler angles and b) using bivectors

We can see that the bivectors do slightly better – important for cases where we would like to predict the state as accurately as possible from the motion data.

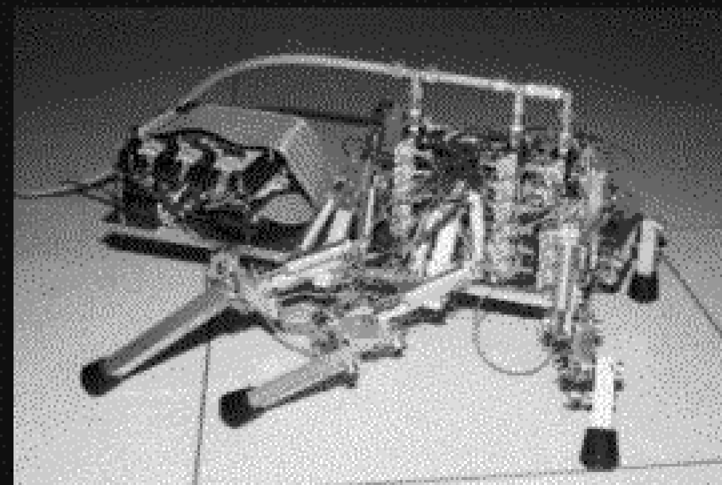
Tracking using bivectors



red = true values blue /green = estimates from bivectors/euler angles

Inverse kinematics

Courtesy CLAWAR website



- Inverse kinematics particularly important in complex machines like this hexapod walking robot

- Given vectors we can recover rotors in two very neat ways:

$$\{e_i\} \xrightarrow{R} \{f_i\} \quad i = 1,2,3 \quad \text{exact}$$

$$\{u_i\} \xrightarrow{R} \{v_i\} \quad i = 1,\ldots,n \quad \text{approx}$$

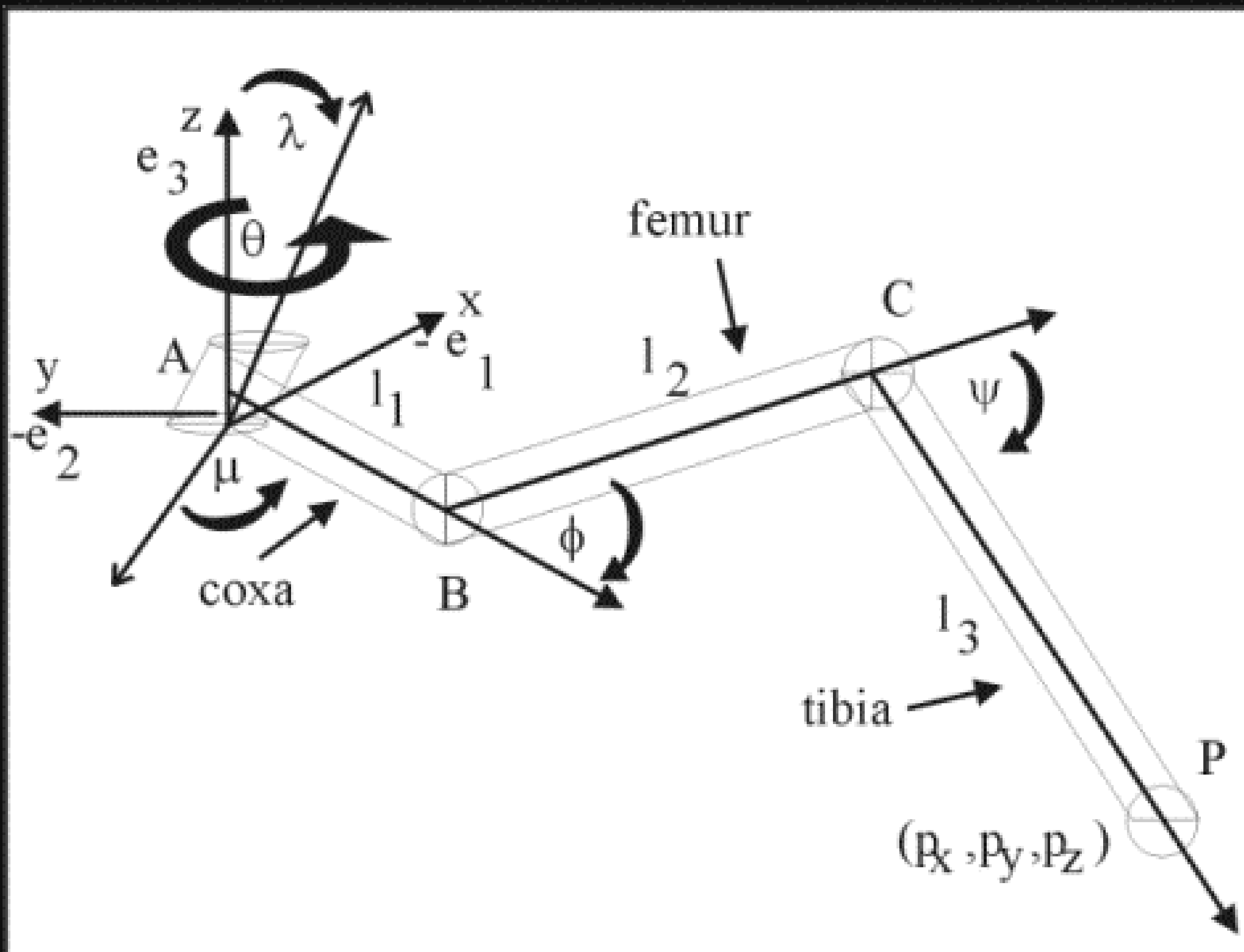
$$R \propto 1 + f_i e^i \quad i = 1,2,3 \quad \text{exact}$$

Rarely any need to resort to coordinates!

$$R \quad \text{satisfies} \quad \sum_{i=1}^{n} R u_i \tilde{R} \wedge v_i = 0 \quad (\text{least squares})$$

An example – simple insect leg

Consider a single hexapod leg shown on previous slide.



Conventionally the forward and inverse kinematics are described in terms of Euler angles.

Simple insect leg: FK & IK

$$
\begin{aligned}
p_x &= (\cos\mu\cos\tilde{\theta} - \cos\lambda\sin\mu\sin\tilde{\theta})[l_2\cos\phi + l_3\cos(\phi+\psi) + l_1] \\
&\quad + \sin\lambda\sin\mu[l_2\sin\phi + l_3\sin(\phi+\psi)] \\
p_y &= (\sin\mu\cos\tilde{\theta} + \cos\lambda\cos\mu\sin\tilde{\theta})[l_2\cos\phi + l_3\cos(\phi+\psi) + l_1] \\
&\quad - \sin\lambda\cos\mu[l_2\sin\phi + l_3\sin(\phi+\psi)] \\
p_z &= \sin\lambda\sin\tilde{\theta}[l_2\cos\phi + l_3\cos(\phi+\psi) + l_1] + \cos\lambda[l_2\sin\phi + l_3\sin(\phi+\psi)]
\end{aligned} \tag{16}
$$

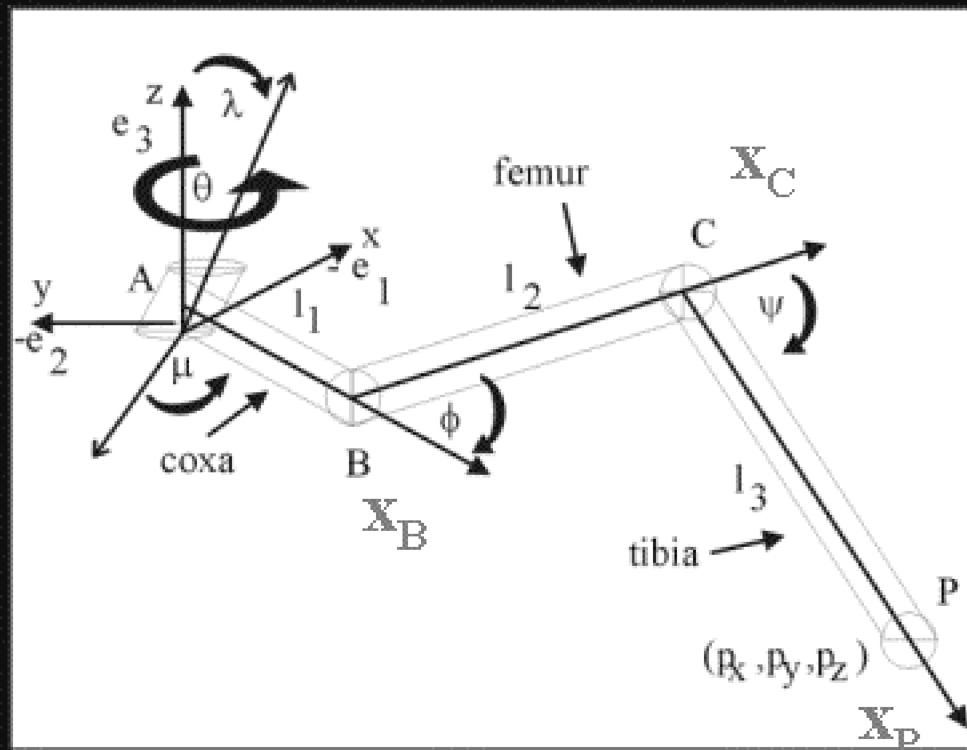
$$
\begin{aligned}
\tilde{\theta} &= \arctan\left(\frac{-p_x\cos\lambda\sin\mu + p_y\cos\lambda\cos\mu + p_z\sin\lambda}{p_x\cos\mu + p_y\sin\mu}\right) \\
\psi &= \arctan\left(-\sqrt{1-\left(\frac{z^2+x^2+y^2-l_2^2-l_3^2}{2l_2l_3}\right)^2} \Big/ \frac{z^2+x^2+y^2-l_2^2-l_3^2}{2l_2l_3}\right) \\
\phi &= \arctan\left(\frac{z}{x^2+y^2}\right) - \arctan\left(\frac{l_3\sin\psi}{l_2+l_3\cos\psi}\right)
\end{aligned} \tag{17}
$$

x,y,z also functions of the angles. Derivation takes many pages of matrix manipulations conventionally

Simple insect leg: FK -- GA

Forward kinematics very easy to write down using rotors as we did for leg:

Establish rotors R_A, R_B, R_C at each of joints and assume rotations from an initially straight position



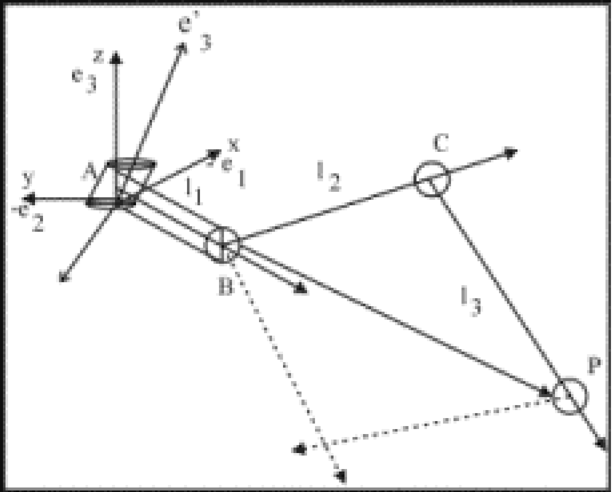
$$x_B = x_A + R_A(-l_1e_2)\tilde{R}_A \equiv R_\mu R_\lambda R_\theta(-l_1e_2)\tilde{R}_\theta\tilde{R}_\lambda\tilde{R}_\mu$$

$$x_C = x_B + R_BR_A(-l_2e_2)\tilde{R}_A\tilde{R}_B = x_B + R_AR'_B(-l_2e_2)\tilde{R}'_B\tilde{R}_A$$

$$x_P = x_C + R_CR_BR_A(-l_3e_2)\tilde{R}_A\tilde{R}_B\tilde{R}_C =$$
$$x_C + R_AR'_BR'_C(-l_3e_2)\tilde{R}'_C\tilde{R}'_B\tilde{R}_A$$

Simple insect leg: IK -- GA

First establish the joint positions via geometry. Then unwrap rotors – 1 page!



$$R_\theta \propto 1 + f^i g_i$$

where $[f_1, f_2, f_3] = [-l_1 e_2, e_3, I f_1 \wedge f_2]$

and $[g_1, g_2, g_3] = [\tilde{R}_\lambda \tilde{R}_\mu (x_B - x_A) R_\mu R_\lambda, e_3, I g_1 \wedge g_2]$

Advantages are not having to deal with any quadrant polarities – no sines or cosines involved in the inversion.

$$R'_B \propto 1 + f^i g_i$$

where $[f_1, f_2, f_3] = [-l_2 e_2, e_1, I f_1 \wedge f_2]$

and $[g_1, g_2, g_3] = [\tilde{R}_A (x_C - x_B) R_A, e_1, I g_1 \wedge g_2]$

$$R'_C \propto 1 + f^i g_i$$

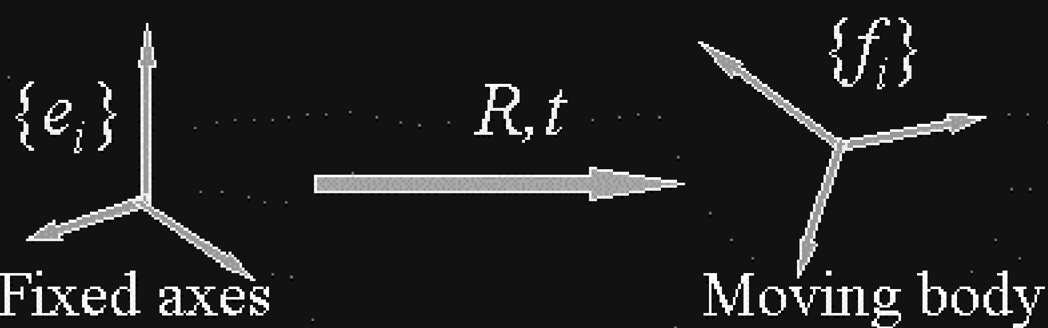
where $[f_1, f_2, f_3] = [-l_3 e_2, e_1, I f_1 \wedge f_2]$

and $[g_1, g_2, g_3] = [\tilde{R}'_B \tilde{R}_A (x_P - x_C) R_A R'_B, e_1, I g_1 \wedge g_2]$

Inverse Dynamics

GA is the perfect tool for cases in which we want to infer dynamical quantities from observed data.

We have been using it to infer forces and torques on a subject during motion from data from the optical motion capture unit – i.e. 3d positions of points on the body over time.



Crucial step is to define motion of body in terms of a time-dependent rotor

$$f_k = R e_k R^\dagger$$

The full dynamical equations can then be written down concisely in terms of the quantities

$$R, \quad \Omega_S, \quad \text{and} \quad \Omega_B = R^\dagger \Omega_S R$$

where Ω_S, Ω_B are the space and body angular velocity bivectors.

Inverse Dynamics

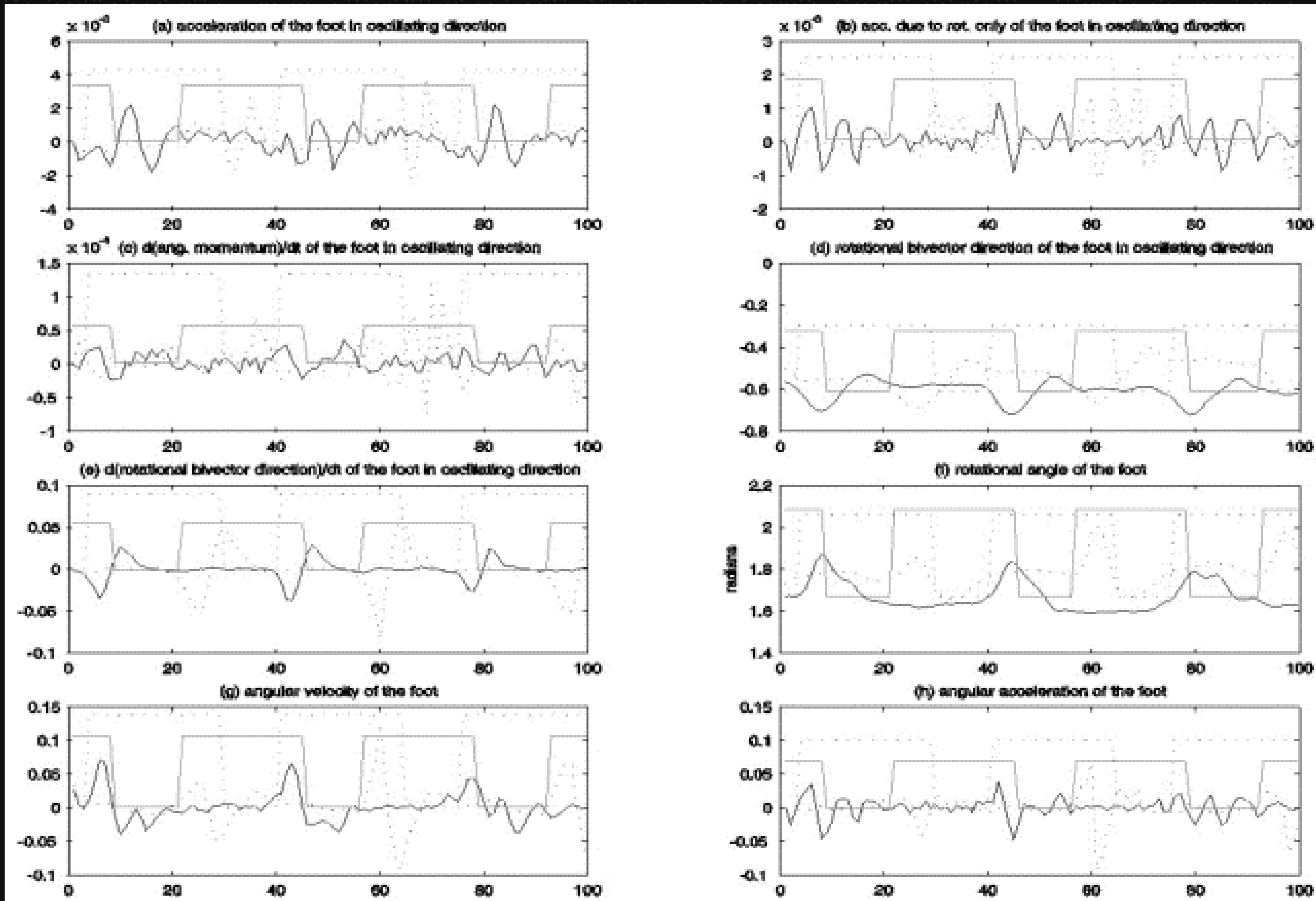
Have a very straightforward system to work with – estimate the bivector B_k $(R_k = e^{-B_k})$ and its first and second derivatives at each time step from the position data, approximate the inertia tensors of limbs, and estimate forces and torques:

In GA the inertia tensor, I, becomes a mapping of bivectors onto bivectors, so that the rate of change of angular momentum of a body can be written as

$$\dot{L} = R[\mathrm{I}(\Omega_B) \times \Omega_B + \mathrm{I}(\dot{\Omega}_B)]R^{\dagger}$$

Graphs on next slide show results of analysing data of person walking on the moving bridge (as shown earlier) – we are able to plot reasonably smooth estimates of accelerations, angles etc in particular directions and to infer forces and torques from these:

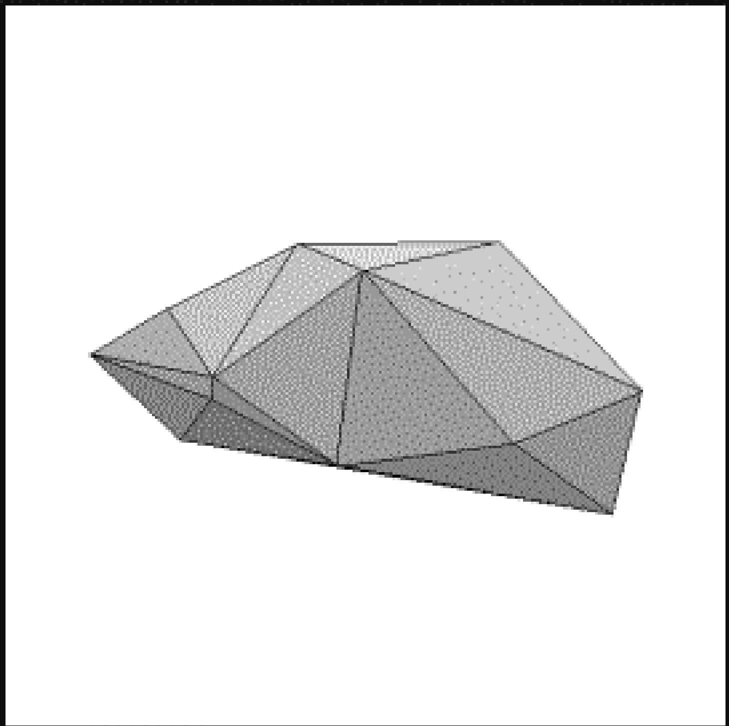
Dynamical data from bridge dataset



A Framework for ray tracing

Given a set of plane reflectors with normals $n_1, n_2, n_3, \ldots$etc and an incoming ray direction m, we form the final output direction after multiple bounces via

Video



$$m' = \pm n_i n_{i-1} \ldots n_1 m n_1 \ldots n_{i-1} n_i$$

This works in the 3d GA and works well for direction, **but doesn't keep track of the position of the rays** and hence (for example) the planes they will intersect in a complex case.

Intersecting rays and planar facets

Use 5d conformal representation of 3d Euclidean space (see later talks). Let Φ be the plane and L be the line we wish to reflect (in 5d these are 4-vectors and 3-vectors resp and e_∞ is a null vector representing the point at infinity)

$$\Phi = e_\infty \wedge A \wedge B \wedge C \wedge D, \quad L = e_\infty \wedge P \wedge Q$$

Find that $L' = \Phi L \Phi$ not only has the correct direction, but ***goes through the point of intersection of the line and the plane.***

Hence we can now carry along positional as well as directional information. Can also reflect in several planes via

$$L' = \Phi_i \Phi_{i-1} \ldots \Phi_1 L \Phi_1 \ldots \Phi_{i-1} \Phi_i$$

output ray will be in the correct place positionally

e

Intersecting rays and planar facets cont..

We can also see easily whether a ray intersects a particular triangular facet in an efficient manner using the idea of ***reciprocal frames in 5d*** – cuts down on the number of conditional operations one needs to do.

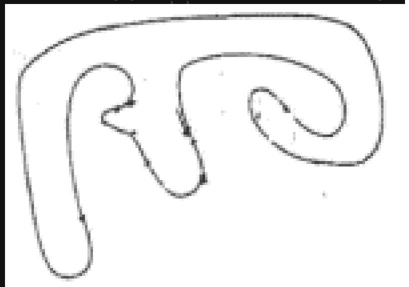
Have been using these ideas in two main applications :

a) Ray tracing – multiple reflections from complex surfaces
b) Surface evolution

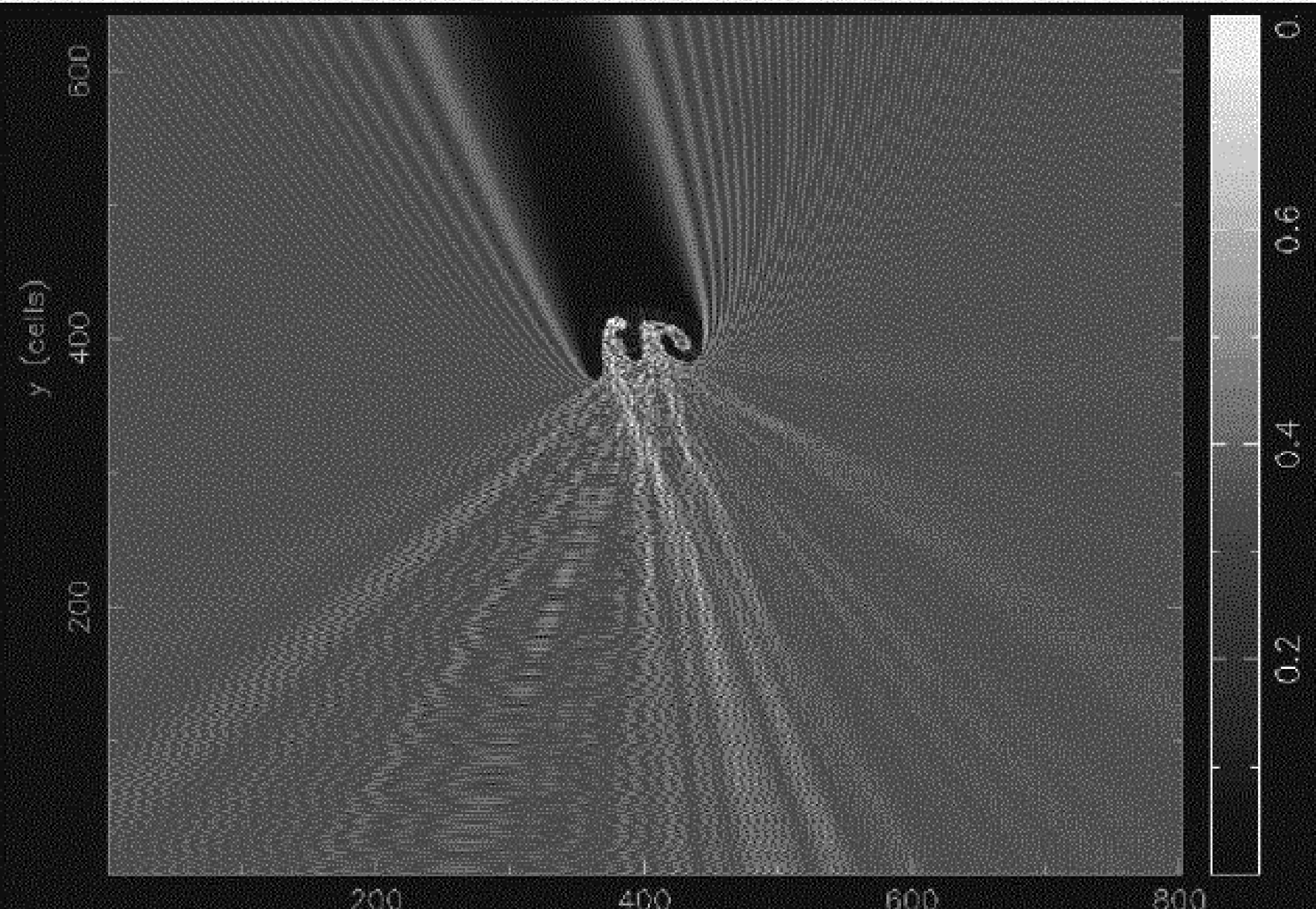
Will show two examples:

1. Scattering of an incident plane wave from a complex 2D shape with concavities – there is little extra computational complexity required in 3d.

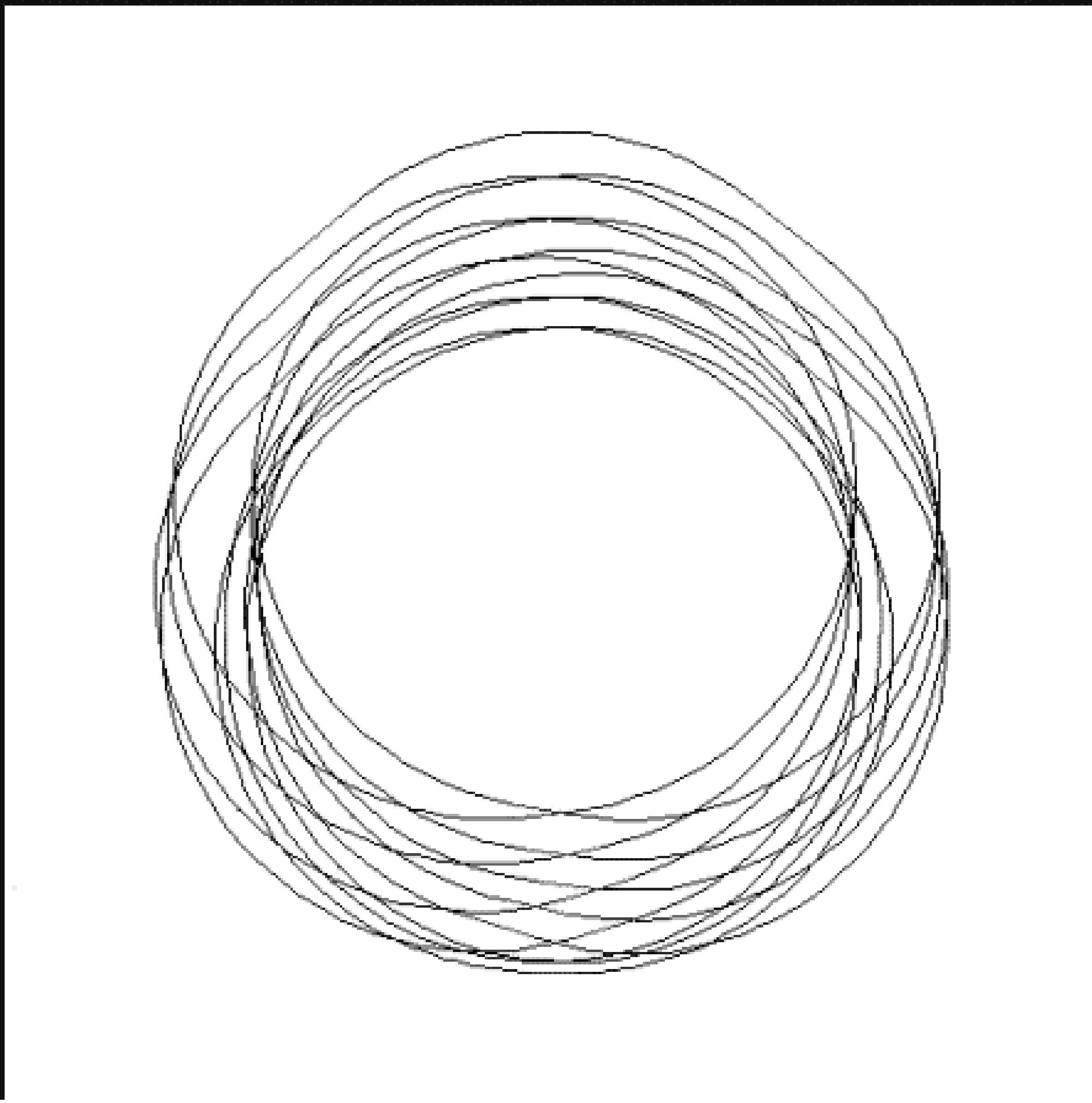
2. Evolving initial circles onto a 2d shape with concavities – again the method remains same when we move to 3d



Surface scattering



Surface evolution



An initially differentiable curve evolves to a shape exhibiting concavities.

Video

Summary

- Have looked at several engineering/computer science applications of GA – have also seen that GA is a very powerful language for physics

- The elegant, generalizable (over dimensions) and efficient way in which GA handles rotations is the key to much of the applications work

- The mathematics involved is considerably simpler than some areas of pure maths (differential topology etc) used in current engineering/comp.sci research.

Geometric Algebra:
Application Studies

Joan Lasenby, Sahan Gamage
and Maurice Ringer
Department of Engineering
University of Cambridge
Trumpington Street
Cambridge, CB2 1PZ, UK
`jl@eng.cam.ac.uk`

Chris Doran
Astrophysics Group
University of Cambridge
Madingley Road
Cambridge, CB3 OHE, UK
`c.doran@mrao.cam.ac.uk`

`http://www-sigproc.eng.cam.ac.uk/vision`
`http://www.mrao.cam.ac.uk/ clifford`

Course Notes to accompany Applications talk: SIGGRAPH 2001, Los Angeles – Inverse kinematics and dynamics

Scope

This set of notes gives some background to the material presented in the applications lectures. More detail can be found in the references listed at the end of these notes.

The notes cover some of the material presented in the lectures, mainly the inverse kinematics and dynamics. The notation follows that in the course although the references listed adopt a slightly different notation scheme.

1 Tracking, analysis and Inverse Kinematics

1.1 Introduction

The main driving force behind the development of the modelling techniques we will describe in subsequent sections has been the need to provide fast and efficient algorithms for optical motion capture. Optical motion capture is a relatively cheap method of producing 3D reconstructions of a subject's motion over time, the results of which can be used in a variety of applications; biomechanics, robotics, medicine, animation etc. Using a system with few cameras (3 or 4) we find that in order to reliably match and track the data (consisting of bright markers placed at strategic points on the subject) we must use realistic models of the possible motion. Once the data has been tracked using such models, we are in a position to analyse the motion in terms of the rotors we have recovered.

The mathematical language we will use throughout will be that of geometric algebra (GA). This language is based on the algebras of Clifford and Grassmann and the form we follow here is that formalised by David Hestenes [1]. There are now many texts and useful introductions to GA, [2, 3, 4, 5].

1.2 Rotations

If, in 3D, we consider a rotation to be made up of two consectutive reflections, one in the plane perpendicular to a unit vector m and the next in the plane perpendicular to a unit vector n, it can easily be shown [4] that we can represent this rotation by a quantity R we call a **rotor** which is given by

$$R = nm$$

Thus a rotor in 3D is made up of a scalar plus a bivector and can be written in one of the following forms

$$R = e^{-B/2} = \exp\left(-I\frac{\theta}{2}n\right) = \cos\frac{\theta}{2} - In\sin\frac{\theta}{2}, \tag{1}$$

which represents a rotation of θ radians about an axis parallel to the unit vector n in a right-handed screw sense. Here the bivector B represents the plane of rotation. Rotors act two-sidedly, ie. if the rotor R takes the vector a to the vector b then

$$b = RaR^{\dagger}.$$

where $R^{\dagger} = mn$ is the reversion of R (i.e the order of multiplication of vectors in any part of the multivector is reversed). We have that rotors must therefore satisfy the constraint that $RR^{\dagger} = 1$. One huge advantage of this formulation is that rotors take the same form, i.e. $R = \pm\exp(B)$ in any dimension (we can define hyperplanes or bivectors in any space) and can rotate any objects, not just vectors; e.g.

$$\begin{aligned} R(a\wedge b)R^{\dagger} &= \langle RabR^{\dagger}\rangle_2 = \langle RaR^{\dagger}RbR^{\dagger}\rangle_2 \\ &= RaR^{\dagger}\wedge RbR^{\dagger} \end{aligned} \tag{2}$$

gives the formula for rotating a bivector.

Before we leave the topic of rotations, we will outline one property of rotors which will turn out to be familiar to us from classical Euler angle descriptions of 3D rotations. Consider an orthonormal basis for 3-space, $\{e_1, e_2, e_3\}$; suppose we perform a rotation R_1, where $R_1 = \mathrm{e}^{-I\theta_1 e_1}$, i.e. we first rotate an angle θ_1 about an axis e_1. We then follow this by a rotation of θ_2 about *the rotated* e_2 axis – this second rotor, R_2, is given by

$$R_2 = \mathrm{e}^{-I\theta_2 R_1 e_2 R_1^{\dagger}}$$

The combined rotation is therefore given by $R_T = R_2 R_1$ – this can be written as follows:

$$\begin{aligned} R_T &= \{\cos\theta_2/2 - IR_1e_2R_1^{\dagger}\sin\theta_2/2\}R_1 \\ &= R_1\{\cos\theta_2/2 - Ie_2\sin\theta_2/2\}R_1^{\dagger}R_1 \\ &= R_1R_2' \end{aligned} \tag{3}$$

since $R_1R_1^{\dagger} = 1$ and $R_1\alpha R_1^{\dagger} = \alpha$ for α a scalar.. Thus if R_2' is the rotation of θ_2 about the *non-rotated* axis (i.e. just e_2 in this case), we see that the compound rotation can be written in two ways

$$R_2R_1 = R_1R_2' \tag{4}$$

Now recall the classical Euler angle formulation: any general rotation can be expressed as follows: a rotation of ϕ about the $\boldsymbol{e}_3$ axis, followed by a rotation of θ about the **rotated** $\boldsymbol{e}_1$ axis, followed by a rotation of ψ about the **rotated** $\boldsymbol{e}_3$ axis [6], as shown in figure 1

Something we always want to do is to apply such a rotation to a vector x. In GA terms we have 3 rotors representing the 3 rotations:

$$R_1 = \exp\{-I\frac{\phi}{2}e_3\},\ R_2 = \exp\{-I\frac{\theta}{2}e_1'\},\ R_3 = \exp\{-I\frac{\psi}{2}e_3''\}$$

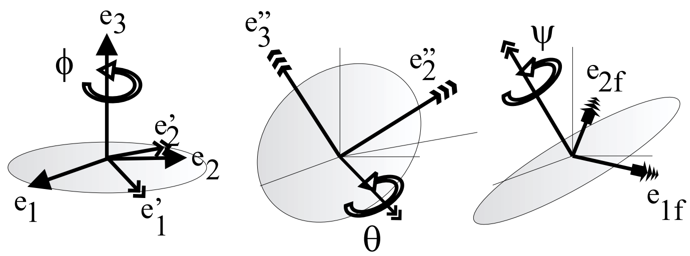


Figure 1: Sketch of the three elementary rotations in the Euler angle formulation – in which initial axes (e_1, e_2, e_3) are rotated to final axes (e_{1f}, e_{2f}, e_{3f})

where $\boldsymbol{e}_1' = R_1 \boldsymbol{e}_1 R_1^\dagger$ and $\boldsymbol{e}_3'' = R_2 R_1 \boldsymbol{e}_3 R_1^\dagger R_2^\dagger$. The combined rotor is

$$R_T = R_3 R_2 R_1 \ \text{ so that } \ x' = R_T x R_T^\dagger$$

This is all very straightforward, mainly because we are dealing with *active* transformations.

Now, if we implement our Euler angle formulation via rotation matrices, [6], we see that we have 3 rotations matrices:

$$A_1 = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}$$

$$A_3 = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

which represent the rotations about the non-rotated axes and we apply these matrices in *reverse* order to form

$$A_T = A_1 A_2 A_3 \qquad \text{so that} \qquad x' = A_T x$$

If R_1', R_2', R_3' are the rotors representing the rotations encoded in A_1, A_2, A_3 (i.e. rotations about the non-rotated axes), then we therefore see that (noting $R_1' = R_1$)

$$R_1' R_2' R_3' = R_3 R_2 R_1$$

which is precisely the formula that we know relates rotations about rotated and non-rotated axes given in equation 4. Confusion often arises due to the *passive* nature of the Euler angle formulation as given in standard textbooks – there is no such confusion possible if we work totally with *active* transformations, as one is forced to do with the rotor formulation.

1.3 Articulated Motion Models: Forward Kinematics and Tracking

We begin by considering a simple model of a leg as two linked rigid rods shown in figure 2. Let us assume that the first rod, AB, can rotate with all degrees of freedom about

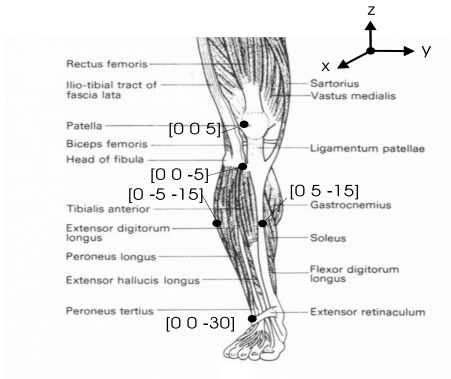


Figure 2: Two linked rigid rods used to simulate the leg

point A but that the second rod, BC, can only rotate in the plane formed by the two rods (i.e. about an axis which is perpendicular to both rods and initially aligned with the e_2 axis). In reality more complex constraints can be considered. e_1, e_2, e_3 form a fixed orthonormal basis oriented as shown. x_a, x_b, x_c are the vectors representing the 3D positions of A, B, C respectively and $x_1 = x_b - x_a$ and $x_2 = x_c - x_b$, which, initially, take the values d_1e_1 and d_2e_1. We can immediately write down the position of points B and C as

$$x_b = x_a + d_1R_1e_1R_1^\dagger \tag{5}$$
$$x_c = x_b + d_2R_2R_1e_1R_1^\dagger R_2^\dagger \tag{6}$$

where we have $R_1 = \exp\{-I\frac{\theta_1}{2}n_1\}$ and we allow for the fact that the point A may move in space (note here that we allow any rotation of rod AB about A, although we may want to only have 2 dof rather than 3 if we are not interested in the orientation of the axes at A). We also have that $R_2 = \exp\{-I\frac{\theta_2}{2}n_2'\}$ with $n_2' = R_1e_2R_1^\dagger$. Using the fact that $R_2R_1 = R_1R_2'$ with $R_2' = \exp\{-I\frac{\theta_2}{2}e_2\}$ we are able to give the position of the ankle, x_c as

$$x_c = x_a + d_1R_1e_1R_1^\dagger + d_2R_1R_2'e_1R^{\dagger\prime}{}_2R_1^\dagger \equiv x_a + R_1\{d_1e_1 + d_2R_2'e_1R^{\dagger\prime}{}_2\}R_1^\dagger \tag{7}$$

Thus we are able to write down, in a manner which deals only with active transformations, such forward kinematics equations for arbitrarily complex mechanisms. But this is not the only advantage of this approach; we can now have the elements of our *state* as *rotors* – it is well known that singularities can occur using Euler angles (i.e. when an angle goes to zero, 90° or other specific ranges) and we can avoid many of these singularities using the rotor components as our variables. The use of such models in optical tracking scenarios is briefly discussed here.

In a typical multi-camera tracking problem where we place markers on joints, the measurements (2D points in the camera planes) will be related to the state via a *measurement equation*:

$$\boldsymbol{y}(k) = H_k(\boldsymbol{x}(k)) + \boldsymbol{w}(k) \tag{8}$$

where the $\boldsymbol{y}(k)$ is our set of measurements (observations) at time $t = k$, $\boldsymbol{x}(k)$ is the state at time $t = k$ (parameters describing our model(s)), and $\boldsymbol{w}(k)$ is a zero-mean random vector representing noise at the detection points. The function H_k relates the model parameters to the observations. In this case we take our model parameters to be the coefficients of the bivectors representing the rotors ($B = b_1 I e_1 + b_2 I e_2 + b_3 I e_3$) and then use expressions such as equation 7 to relate these to our observations.

The *process* equation

$$\boldsymbol{x}(k+1) = F_k(\boldsymbol{x}(k)) + \boldsymbol{v}(k)$$

tells us how our system (model) evolves in time; here $\boldsymbol{v}(k)$ represents the process noise. In the case described, F_k tells us how we believe the bivectors to be evolving – one might argue that the variation of the bivectors will be smoother than the evolution of separate Euler-angles.

In general, H_k will be extremely non-linear and so the above problem can be solved by applying an extended Kalman filter (EKF) to update our model estimates and predicted observations at each time step.

A detailed comparison of the difference between using Euler-angles and using bivector coefficients as the scalar model parameters in such tracking problems will be given elsewhere.

1.4 Inverse kinematics (IK)

Inverse kinematics is the procedure of recovering the model or state parameters given the measurements – in particular, when incomplete sets of measurements are given (i.e. not all the joint coordinates) we can, in certain cases, recover a unique model or a specified family of solutions. In this section we shall outline the use of GA in solving IK problems by consideration of a particular, fairly simple, example. The example we choose is the following (it is one which often appears in standard texts); a system consisting of three linked rigid rods representing a typical insect leg – such a setup is commonly used in walking robots and is illustrated in figure 3. Here we fix a set of axes represented by unit vectors (e_1, e_2, e_3) (note that in the figure, $-e_1, -e_2, e_3$ are shown) at the basal joint, so that the angle of the first link, or coxa, is given by the Euler angles (θ, λ, μ), and the rotor representing this rotation is

$$R_A = \mathrm{e}^{-I\frac{\mu}{2}e_3}\mathrm{e}^{-I\frac{\lambda}{2}e_1}\mathrm{e}^{-I\frac{\theta}{2}e_3} \equiv R_\mu R_\lambda R_\theta \tag{9}$$

Generally the angles (λ, μ) are taken as known, so that θ alone describes the position of the first link. The second (femur) and third (tibia) links are such that only rotation in the plane of the three links is allowed, so that the positions of the leg are fully described by

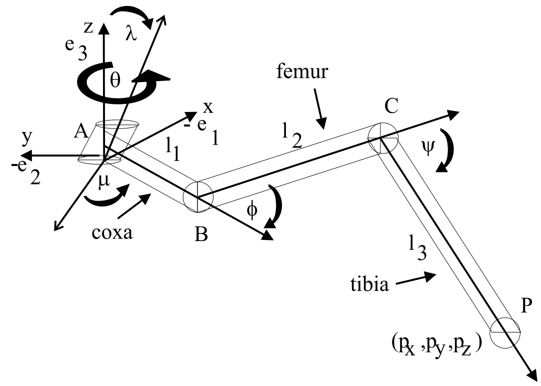


Figure 3: Three linked rigid rods representing the leg of an insect

two further angles, ϕ and ψ as shown in figure 3. If we take our initial configuration to be that in which the leg is fully extended with all links lying along the rotated (by R_A) e_2 direction (i.e. ϕ and $\psi = 0$), then the rotations at joints B and C are given by

$$R_B = \mathrm{e}^{I\frac{\phi}{2}e'_1} \quad \text{and} \quad R_C = \mathrm{e}^{I\frac{\psi}{2}e''_1} \tag{10}$$

where $e'_1 = R_A e_1 R_A^\dagger$, $e''_1 = R_B R_A e_1 R_A^\dagger R_B^\dagger$. Note here that we are rotating about the $-e_1$ direction in order to give the sense of ϕ and ψ shown in figure 3. We are thus able to write down the postion vectors of all joints and finally of the foot position x_P as follows

$$\begin{aligned} x_B &= x_A + R_A(-l_1 e_2)R_A^\dagger \equiv R_\mu R_\lambda R_\theta(-l_1 e_2)R_\theta^\dagger R_\lambda^\dagger R_\mu^\dagger && (11)\\ x_C &= x_B + R_B R_A(-l_2 e_2)R_A^\dagger R_B^\dagger = x_B + R_A R'_B(-l_2 e_2)R^{\dagger\prime}{}_B R_A^\dagger && (12)\\ x_P &= x_C + R_C R_B R_A(-l_3 e_2)R_A^\dagger R_B^\dagger R_C^\dagger = \\ & \quad x_C + R_A R'_B R'_C(-l_3 e_2)R^{\dagger\prime}{}_C R^{\dagger\prime}{}_B R_A^\dagger && (13) \end{aligned}$$

where $R'_B = \mathrm{e}^{I\frac{\phi}{2}e_1}$ and $R'_C = \mathrm{e}^{I\frac{\psi}{2}e_1}$. We can therefore write x_P as

$$x_P = x_A + R_A\{-l_1 e_2 + R'_B\{-l_2 e_2 + R'_C(-l_3 e_2)R^{\dagger\prime}{}_C\}R^{\dagger\prime}{}_B\}R_A^\dagger \tag{14}$$

This uniquely gives the forward kinematic equations in terms of the three rotors R_A, R_B, R_C; if one was to convert this to angles one gets the following equations (which are conventionally obtained when one uses transformation matrices to denote position of one joint relative to the previous joint [7]):

$$\begin{aligned} p_x &= (\cos\mu\cos\tilde{\theta} - \cos\lambda\sin\mu\sin\tilde{\theta})[l_2\cos\phi + l_3\cos(\phi+\psi) + l_1] \\ &+ \sin\lambda\sin\mu[l_2\sin\phi + l_3\sin(\phi+\psi)] \\ p_y &= (\sin\mu\cos\tilde{\theta} + \cos\lambda\cos\mu\sin\tilde{\theta})[l_2\cos\phi + l_3\cos(\phi+\psi) + l_1] \qquad (15)\\ &- \sin\lambda\cos\mu[l_2\sin\phi + l_3\sin(\phi+\psi)] \\ p_z &= \sin\lambda\sin\tilde{\theta}[l_2\cos\phi + l_3\cos(\phi+\psi) + l_1] + \cos\lambda[l_2\sin\phi + l_3\sin(\phi+\psi)] \end{aligned}$$

In the above, $\tilde{\theta} = \theta - \pi/2$, since the convention (following Denavit-Hartenberg) is to measure this basal rotation angle from the e_2 axis rather than from the e_1 axis as our rotor formulation has done. Now, the inverse kinematics comes in when we try to recover the joint angles $(\tilde{\theta}, \phi, \psi)$ given (p_x, p_y, p_z) (and the origin of coordinates). Conventionally the solution is obtained by a series of fairly involved matrix manipulations to give the following expressions for the joint angles:

$$\begin{aligned}
\tilde{\theta} &= \arctan\left(\frac{-p_x\cos\lambda\sin\mu + p_y\cos\lambda\cos\mu + p_z\sin\lambda}{p_x\cos\mu + p_y\sin\mu}\right) \\
\psi &= \arctan\left(-\sqrt{1 - \left(\frac{z^2+x^2+y^2-l_2^2-l_3^2}{2l_2l_3}\right)^2} \Big/ \frac{z^2+x^2+y^2-l_2^2-l_3^2}{2l_2l_3}\right) \\
\phi &= \arctan\left(\frac{z}{x^2+y^2}\right) - \arctan\left(\frac{l_3\sin\psi}{l_2+l_3\cos\psi}\right) \qquad (16)
\end{aligned}$$

where

$$x = p_x\cos\mu + p_y\sin\mu - l_1\cos\tilde{\theta} \qquad (17)$$

$$y = -p_x\cos\lambda\sin\mu + p_y\cos\lambda\cos\mu + p_z\sin\lambda - l_1\sin\tilde{\theta} \qquad (18)$$

$$z = p_x\sin\lambda\sin\mu - p_y\sin\lambda\cos\mu + p_z\cos\lambda \qquad (19)$$

In standard texts it is often noted that it is better to express joint angles in terms of arctangent functions to avoid quadrant polarities – we will return to this point later when discussing problems with this Euler angle formulation. Suppose that we have the points x_a, x_b, x_c, x_p, we will now show that it is straightforward, from equations 11-13, to recover each of the rotors, R_A, R_B, R_C. In order to do this we shall use a simple result from GA (see [4] for more details). Suppose that a set of three (non-coplanar and not necessarily orthonormal) unit vectors e_1, e_2, e_3 is rotated by a rotor R into a set of three other (necessarily non-coplanar) unit vectors f_1, f_2, f_3 – then the unique rotor which performs this job is given by

$$R \propto 1 + e^i f_i \qquad (20)$$

where the proportionality factor is easily found by ensuring $R\tilde{R} = 1$ and $\{e^i\}$ denotes the reciprocal frame of $\{e_i\}$. The reciprocal frame $\{e^i\}$ is such that $e^i \cdot e_j = \delta^i_j$ and can be formed (for 3D) as follows

$$\begin{aligned}
e^1 &= \frac{1}{\alpha} I e_2 \wedge e_3 \\
e^2 &= \frac{1}{\alpha} I e_3 \wedge e_1 \qquad (21) \\
e^3 &= \frac{1}{\alpha} I e_1 \wedge e_2, \qquad (22)
\end{aligned}$$

where $I\alpha = e_3 \wedge e_2 \wedge e_1$.

This provides us with a remarkably easy way of extracting rotors if we know the joint coordinates. Let us first consider equation 11 for R_A. We can rewrite this equation as

$$R_\lambda^\dagger R_\mu^\dagger (x_B - x_A) R_\mu R_\lambda = R_\theta(-l_1 e_2) R_\theta^\dagger \qquad (23)$$

From this we can see that the vector $f_1 = -l_1 e_2$ is rotated into the vector $g_1 = R_\lambda^\dagger R_\mu^\dagger (x_B - x_A) R_\mu R_\lambda$ and also that, since $R_\theta = \mathrm{e}^{-I\frac{1}{2}\theta e_3}$, the vector $f_2 = e_3$ is rotated into itself, i.e. $g_2 = e_3$. From this it follows that $f_3 = I f_1 \wedge f_2$ must be rotated into $g_3 = I g_1 \wedge g_2$. Thus, using equations 22 we can form $\{f^i\}$ and the rotor R_θ as follows

$$\begin{aligned} R_\theta &\propto 1 + f^i g_i \\ \text{where} \quad & [f_1, f_2, f_3] = [-l_1 e_2, e_3, I f_1 \wedge f_2] \\ \text{and} \quad & [g_1, g_2, g_3] = [R_\lambda^\dagger R_\mu^\dagger (x_B - x_A) R_\mu R_\lambda, e_3, I g_1 \wedge g_2] \end{aligned} \tag{24}$$

Thus R_A is then recovered from equation 9. Using this we can now look at equation 12 which can be rewritten as

$$R_A^\dagger (x_C - x_B) R_A = R'_B (-l_2 e_2) R^{\dagger\prime}_B \tag{25}$$

We can then invert as above to give

$$\begin{aligned} R'_B &\propto 1 + f^i g_i \\ \text{where} \quad & [f_1, f_2, f_3] = [-l_2 e_2, e_1, I f_1 \wedge f_2] \\ \text{and} \quad & [g_1, g_2, g_3] = [R_A^\dagger (x_C - x_B) R_A, e_1, I g_1 \wedge g_2] \end{aligned} \tag{26}$$

Finally, R'_C can be recovered by precisely the same means using

$$\begin{aligned} R'_C &\propto 1 + f^i g_i \\ \text{where} \quad & [f_1, f_2, f_3] = [-l_3 e_2, e_1, I f_1 \wedge f_2] \\ \text{and} \quad & [g_1, g_2, g_3] = [R^{\dagger\prime}_B R_A^\dagger (x_P - x_C) R_A R'_B, e_1, I g_1 \wedge g_2] \end{aligned} \tag{27}$$

Thus, we see that we are able to invert our forward kinematic equations trivially if we have the coordinates of the joints. Of course, the IK problem as we described it involved being given only x_A and x_P. The plan we advocate is therefore to find x_B and x_C by purely geometric means as an initial stage, followed by the rotor inversion process described above. To illustrate this, consider how we would find x_B, x_C for the given example.

Taking x_A at the origin, we know that e'_3 and x_P must define the plane in which all the links must lie, call this plane Φ – see figure 4. We can form x_B via

$$x_B = l_1 \frac{(x_p - (x_P \cdot e'_3) e'_3)}{|x_p - (x_P \cdot e'_3) e'_3|} \tag{28}$$

There are clearly two possibilities for x_C, given by the intersections of the circles lying in the plane Φ having centres and radii given by (x_B, l_2) and (x_P, l_3). If we then define $e_{||} = (x_P - x_B)/(|x_P - x_B)|$ and $e_\perp$ a vector perpendicular to $e_{||}$ lying in Φ, it is not hard to show that x_C is given by

$$x_C = x_{||} e_{||} + x_\perp e_\perp$$

where

$$x_{||} = \frac{l_2^2 - l_3^2 + (x_P - x_B)^2}{2|x_P - x_B|} \tag{29}$$

$$x_\perp^2 = -\left\{ \frac{[(l_2 - l_3)^2 - (x_P - x_B)^2][(l_2 + l_3)^2 - (x_P - x_B)^2]}{4|x_P - x_B|} \right\} \tag{30}$$

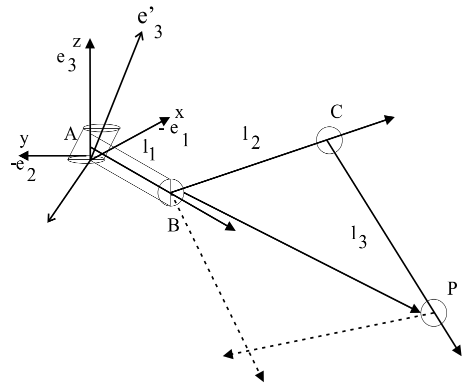


Figure 4: Figure illustrating setup used to determine joint positions from geometry

When the geometry is more complex than given in this example (indeed, things will get more complicated if we also have prismatic joints rather than simple revolute joints) the joint positions, or family of joint positions are found by intersecting circles, spheres, planes and lines (with possible dilations) in 3D. The system that we are currently working on performs this initial geometric stage in the 5D conformal geometric algebra [8, 9]. This framework provides a very elegant means of dealing with incidence geometry and extends the functionality of projective geometry to include circles and spheres. A feature of the conformal setting is that rotations, translations, dilations and inversions in 3D all become rotors in 5D.

We now return to the question of whether we gain any advantages from doing our IK problems in geometric algebra. In the simple case illustrated, simulations have shown that we can recover the rotors (there always exist two sets of solutions) exactly for any combination of angles – there is no need to restrict any of the angles to particular ranges. However, when the equations in 16 are used to recover angles, we find that the whole process is plagued with conditionals, i.e. the correct solutions are obtained only if signs of various terms are checked for various angles in various ranges. From a computing point of view this is expensive and may ultimately lead to hard-to-track-down errors.

1.5 Conclusions

Here we have illustrated how the geometric algebra, and particularly the rotor formulation within the algebra, can be used as a mathematical system in which forward kinematics, motion modelling and inverse kinematics can be elegantly expressed. The formulations given have been put to use in tracking problems in which optical motion capture data is tracked via constrained articulated models and in inverse kinematics of simple leg structures. We believe that the system as outlined here has enormous potential in more complex inverse kinematics problems where we would like to define families of possible solutions – the key here would be to do the initial geometry stage via a 5D conformal geometric algebra. Work in progress also includes the analysis of human motion data via our articulated models in an attempt to understand how motions are described using the rotor formulation.

2 Inferring dynamical information from 3D position data using geometric algebra

2.1 Introduction

Estimating inverse dynamic (ID) quantities is essential in areas such as robot control, biomedical engineering and animation. In the field of robotics there are numerous techniques and procedures for calculating these quantities [10, 11]. The computational procedure given in [11] for estimating ID quantities is the Luh-Walker-Paul algorithm [12]. Also in the context of estimating ID quantities from marker data, [13] has presented an inertial model and a method to calculate the joint moments, although this is not an explicit algorithm for calculation of these quantities. Here we present a step-by-step algorithm to estimate the ID quantities using only the 3D positions of markers attached to an articulated body in general motion using Geometric Algebra (GA). The simplicity of the derivations given here is due to the fact that the rotation of a body is represented as a single quantity; namely the rotational bivector. When Euler angles are used to model the rotation it is extremely difficult to formulate the ID quantities in a simple manner. Using the standard technique of employing angular velocity vectors [11, 13] does not yield a simple connection between the rotational quantities and the actual ID parameters. In such formulations it is not clear how to obtain the angular velocity vector in numerical calculations when the axis and the angle of rotation change with time. Since the direction and the magnitude of rotation are incorporated into a single quantity when rotational bivectors are used, each kinematic and dynamic parameter can be expressed directly.

First we present the basic formulation of the dynamic equations in GA and then derive the angular velocity and acceleration bivectors given the rotations. Then we adapt these results for calculations with marker data. We apply the techniques to real world data obtained from an experimental setup of a person walking on a moving bridge.

2.2 Some Basic Formulations

Here we either derive or state some basic formulae needed for the calculation of inverse dynamics.

2.3 Angular Velocity

If set of vectors $\{f_k\}$ on a body rotating in space can be related to a fixed time independent set of corresponding vectors $\{e_k\}$ by a time dependent rotor R, we can write

$$f_k = Re_kR^\dagger. \tag{31}$$

Define the angular velocity bivector of the rotating system [14, 15] in space, Ω_S, via the equation

$$\dot{f}_k = -\Omega_S \times f_k \tag{32}$$

where operator $\times$ denotes the commutator product defined as $A\times B = -B\times A = (AB - BA)/2$ and $\dot{f}_k$ are the velocity vectors. This is analogous to the 'conventional' definition of $\dot{f}_k = \omega \mathbf{x} f_k$ where **x** denote the vector cross product and ω is the angular velocity vector which is related to Ω_S by $\Omega_S = I\omega$ [14] with I being the pseudoscalar in 3D. Equation (31) can be differentiated with respect to (wrt) time to give

$$\dot{f}_k = \dot{R}e_k R^\dagger + Re_k \dot{R}^\dagger. \tag{33}$$

Note that $\dot{e}_k = 0$ since e_k is fixed. But since $RR^\dagger = 1$, $\dot{R}R^\dagger + R\dot{R}^\dagger = 0$, and therefore

$$\dot{R}^\dagger = -R^\dagger \dot{R} R^\dagger. \tag{34}$$

Equation (33) can then be re-arranged as $\dot{f}_k = \left(2\dot{R}R^\dagger\right)\times f_k$. It can be proved [14] that the quantity $\dot{R}R^\dagger$ is a bivector. Hence we can associate it with the angular velocity bivector of equation (32);

$$\Omega_S = -2\dot{R}R^\dagger \tag{35}$$

and thus $\dot{f}_k = f_k \times \Omega_S$. It is also possible to rearrange equation (35) to give

$$\dot{R} = -\frac{1}{2}\Omega_S R. \tag{36}$$

The angular velocity referred back to the body, Ω_B, is the 'body' angular velocity and is defined [14, 15] by

$$\Omega_S = R\Omega_B R^\dagger. \tag{37}$$

2.3.1 Linear Velocity, Acceleration and inertial force

In general the points on a rigid body which is in general motion relative to a measuring coordinate system can be expressed as

$$\mathbf{y}_i = R\mathbf{x}_i R^\dagger + \mathbf{d} \tag{38}$$

where $\mathbf{y}_i$ and $\mathbf{d}$ are the ith point and the displacement of the Centre of Mass (CoM) of the body in the observation frame respectively. $\mathbf{x}_i$ is the ith point referred to a conveniently chosen fixed set of axes on the body placed at the CoM. Hence, $\mathbf{x}_i$ has no time dependence assuming rigidity of the body. Differentiating equation (38) wrt to time gives the velocity of the point on the body in the observation frame $\dot{\mathbf{y}}_i$ as

$$\dot{\mathbf{y}}_i = \dot{R}\mathbf{x}_i R^\dagger + R\mathbf{x}_i \dot{R}^\dagger + \dot{\mathbf{d}}.$$

Using equations (34) and (35) it is possible to derive (see for e.g. [16]),

$$\dot{\mathbf{y}}_i = (\mathbf{y}_i - \mathbf{d}) \times \Omega_S + \dot{\mathbf{d}} \equiv R\mathbf{x}_i R^\dagger \times \Omega_S + \dot{\mathbf{d}}. \tag{39}$$

Differentiating equation (39) wrt to time again and substituting for $(\dot{\mathbf{y}}_i - \dot{\mathbf{d}})$ gives the acceleration as

$$\ddot{\mathbf{y}}_i = (\mathbf{y}_i - \mathbf{d}) \times \dot{\Omega}_S + [(\mathbf{y}_i - \mathbf{d}) \times \Omega_S] \times \Omega_S + \ddot{\mathbf{d}}.$$

Hence using Newton's second law, the inertial force, F, acting on the body can be written as

$$F = m\ddot{\mathbf{y}}_i$$

where m is the mass of the body if the observation frame is an inertial (constant velocity) frame of reference.

2.3.2 Angular momentum, inertia tensor and inertial torque

It is straightforward to derive the angular momentum bivector, L, of a body. In [14] it is shown to be given by

$$L = R\mathcal{I}(\Omega_B)R^{\dagger} \quad \text{with} \quad \mathcal{I}(B) = \int d^3x\rho\mathbf{x}\wedge(\mathbf{x}\cdot B) \tag{40}$$

where $\mathcal{I}(.)$ is a linear mapping of bivectors onto bivectors, and is the inertia tensor of the body. Since the inertia tensor is an integration referred to the 'fixed' copy vectors, it is time independent. But note that Ω_B is time dependent. Inertial torque, τ, satisfies $\tau = \dot{L}$ and we can form $\dot{L}$ by differentiating equation (40) wrt time to give

$$\dot{L} = \dot{R}\mathcal{I}(\Omega_B)R^{\dagger} + R\mathcal{I}(\Omega_B)\dot{R}^{\dagger} + R\mathcal{I}(\dot{\Omega}_B)R^{\dagger}.$$

But using equation (34) $\dot{L}$ can be expressed ([14, 16]) as

$$\dot{L} = -\frac{1}{2}\Omega_S R\mathcal{I}(\Omega_B)R^{\dagger} + \frac{1}{2}R\mathcal{I}(\Omega_B)R^{\dagger}\Omega_S + R\mathcal{I}(\dot{\Omega}_B)R^{\dagger}.$$

Hence from equation (37), inertial torque is given as ([14, 16]),

$$\tau = \dot{L} = R\left[\mathcal{I}(\Omega_B)\times\Omega_B + \mathcal{I}(\dot{\Omega}_B)\right]R^{\dagger}. \tag{41}$$

2.4 Calculations in terms of rotational bivectors

Here we present a method to calculate the angular velocity bivector Ω_S and the angular acceleration bivector $\dot{\Omega}_S$ given only the rotational bivector B. The derivation of Ω_S is similar to the method presented in [17]. First we use the definition of the rotor R in terms of the bivector B

$$R \equiv \exp(-B) \equiv \exp(-\frac{\theta}{2}\hat{B}) \equiv \cos(\theta/2) - \sin(\theta/2)\hat{B} \tag{42}$$

where $B = \frac{\theta}{2}\hat{B}$ and $\hat{B}^2 = -1$, to evaluate

$$\dot{R} = -\frac{\dot{\theta}}{2}\sin(\theta/2) - \frac{\dot{\theta}}{2}\cos(\theta/2)\hat{B} - \sin(\theta/2)\dot{\hat{B}} = -\frac{\dot{\theta}}{2}\hat{B}R - \sin(\theta/2)\dot{\hat{B}}.$$

Hence we can write $\Omega_S = -2\dot{R}R^{\dagger}$ in terms of the above as

$$\Omega_S = \dot{\theta}\hat{B} + 2\sin(\theta/2)\dot{\hat{B}}R^{\dagger}. \tag{43}$$

Since it is easier to evaluate $\dot{B}$ than $\dot{\hat{B}}$ and $\dot{\theta}$, it makes sense to write $\dot{\theta}$ and $\dot{\hat{B}}$ in terms of $\dot{B}$

$$\theta^2 = -4BB,\, 2\theta\dot{\theta} = -4\dot{B}B - 4B\dot{B},\, \dot{\theta} = -4\left(\dot{B}\underline{\times}B\right)/\theta \tag{44}$$

where $A\underline{\times}B = (AB + BA)/2$ is the anti-commutator product. Since $\hat{B} = 2B/\theta$ and therefore $\theta\hat{B} = 2B$, we have

$$\dot{\theta}\hat{B} + \theta\dot{\hat{B}} = 2\dot{B} \implies \dot{\hat{B}} = \left(2\dot{B} - \dot{\theta}\hat{B}\right)/\theta. \tag{45}$$

Note that the formulae for $\dot\theta$ and $\dot{\hat{B}}$ in [17] are incorrect due to an error in defining equation (42): in [17] B is taken as $\theta\hat{B}$ rather than $\frac{\theta}{2}\hat{B}$. All subsequent derivations in [17] are correct if we substitute $|B|$ for θ.

In order to evaluate the angular acceleration, $\dot\Omega_S$, in terms of θ, B, $\dot B$ and $\ddot B$, we differentiate equation (43) wrt time and substitute from equation (34) for $\dot{R}^\dagger$ to give

$$\dot\Omega_S = \ddot\theta\hat{B} + \dot\theta\dot{\hat{B}} + \dot\theta\cos(\theta/2)\dot{\hat{B}}R^\dagger + 2\sin(\theta/2)\ddot{\hat{B}}\tilde{R} + \sin(\theta/2)\dot{\hat{B}}R^\dagger\Omega_S \tag{46}$$

Here, $\ddot\theta$ and $\ddot{\hat{B}}$ are to be evaluated. Differentiating equations (44) and (45) gives

$$\ddot\theta = -\left[4\left(\ddot{B}\underline{\times}B\right) + 4\left(\dot{B}\right)^2 + \dot\theta^2\right]/\theta \quad \text{and} \quad \ddot{\hat{B}} = \left(2\ddot{B} - \ddot\theta\hat{B} - 2\dot\theta\dot{\hat{B}}\right)/\theta$$

Also via equation (37), it is possible to derive the relationship between $\dot\Omega_B$ and $\dot\Omega_S$ as

$$\dot\Omega_B = R^\dagger\dot\Omega_S R. \tag{47}$$

A complete derivation of all the above basic results can be found in [16].

Also, as a first approximation, we use the two sided Euler formulae for the numerical derivatives but if higher accuracy is needed, especially in the noisy data case, a polynomial fit for the function around each data point can be used. A sophisticated realisation of this is the Savitzky-Golay filters as implemented in [18].

3 Algorithm for Inverse Dynamics

Here we describe an algorithm for calculating inertial forces and torques given only the marker positions attached to an articulated model using results from previous sections. The assumptions are;

1. the 3D marker coordinates (possibly noisy) at each time instance are given
2. the time intervals for the data sets are known or constant
3. markers are labelled in the sense that it is known a priori to which link a marker is attached
4. each link can be modelled as a rigid body, attached either to a ball or a hinge joint.
5. the principal axes of inertia for each link in the observation coordinate frame are known
6. the centre of mass (CoM) in relation to the joint location is known.

Hence each marker position can be expressed as

$$R_k(l)\mathbf{e}_r^p(l)R_k^\dagger(l) + \mathbf{t}_k = \mathbf{v}_k^p(l) \tag{48}$$

where $R_k(l)$ is the rotation at the kth time instance of the lth link relative to a given observation reference coordinate frame. Also $\mathbf{e}_r^p(l)$ is the vector from the joint to the pth marker at the rth time instance (usually we take $r = 1$), $\mathbf{t}_k$ is the translation vector of the joint relative to the observation reference frame.

Under the assumptions given above, the inertial forces and torques can be calculated by the following algorithm;

1. estimate the joint locations $\mathbf{t}_k$, for all time instances
2. estimate the link rotations, $R_k(l)$, relative to the coordinate system at the first time instance, for all time instances and for all links
3. calculate the vector from a joint to its CoM, $\mathbf{c}_k(l)$, for all time instances.
4. calculate the total rotation from the principal axes of inertia to the current observation, $R_k^T(l)$, for each link at each time instance
5. calculate the corresponding rotational bivector $B_k^T(l)$ at each time instance and for each link
6. estimate the $\dot{B}_k^T(l)$ and hence calculate inertial forces and torques.

Using the techniques described in [19], it is possible to estimate $\mathbf{t}_k$ and $R_k(l)$ in a least squares sense. When there is a hierarchical kinematic chain, the *local averaging global* method in [19] can be used. In that method the positions of the joints are built up from considering only one joint at a time and averaging the results over the common links between joints.

Then the position vector from the CoR in the case of a ball joint and the perpendicular distance from the AoR in the case of a hinged joint, $\mathbf{c}_k(l)$, can be calculated if the relative location of the CoM to the joint is known (e.g. middle of the link).

If the rotation required to bring the principal axes of inertia for each link to the observation frame, $r_r(l)$, is known (assumption 5), the total rotation of the principal axes can be expressed as

$$R_k^T(l) = R_k(l) r_r(l).$$

Hence $B_k^T(l)$ can be estimated from

$$R_k^T(l) = \exp\left(-B_k^T(l)\right). \tag{49}$$

3.1 Dynamical Equilibrium in the model

In a system that has one or several articulated chains connected to a central body, equilibrium of the central body can be calculated by working from the bottom of the articulated hierarchy and transferring forces up to the central body. This is also the foundation of Luh-Walker-Paul's algorithm [12]. Considering the free body diagram for a single link the forces acting on the link can be expressed as

$$\mathbf{F}_k^v(l) - \mathbf{F}_k^v(l+1) + m(l)\mathbf{g} = m(l)\ddot{\mathbf{y}}_k^v(l) \tag{50}$$

where $\mathbf{F}_k^v(l)$ is the force vector acting *on* the joint at the beginning of the link l on the vth kinematic chain at the kth time instance, $\mathbf{F}_k^v(l+1)$ is the force vector acting *on* the joint at the beginning of the link $l+1$ on the vth kinematic chain at the kth time instance, $m(l)$ the mass of the link l, $\mathbf{g}$ the gravitational pull per unit mass and $\ddot{\mathbf{y}}_k^v(l)$ is the acceleration of the CoM. Writing the set of equations for the link L_v down to link 1 where L_v is the last link of the vth chain, we have

$$\mathbf{F}_k^v(1) = \mathbf{F}_k^v(L_v+1) + \sum_{l=1}^{L_v} m(l)\ddot{\mathbf{y}}_k^v(l) - \sum_{l=1}^{L_v} m(l)\mathbf{g} \tag{51}$$

where $-\mathbf{F}_k^v(L_v+1)$ is taken to be the external force acting at the end of link L_v. Considering the equilibrium of the central body gives

$$-\sum_{v=1}^{V} \mathbf{F}_k^v(1) = m(b)\ddot{\mathbf{y}}_k^b \tag{52}$$

assuming V chains are connected to the central body b. Hence substituting for $\mathbf{F}_k^v(1)$ in equation (52) from equation (51) gives

$$\sum_{v=1}^{V} \mathbf{F}_k^v(L_v+1) = \sum_{v=1}^{V}\sum_{l=1}^{L_v} m_v(l)\mathbf{g} - \sum_{v=1}^{V}\sum_{l=1}^{L_v} m_v(l)\dot{\mathrm{d}}\mathbf{oty}_k^v(l) - m(b)\ddot{\mathbf{y}}_k^b \tag{53}$$

An analogous torque relationship can be derived by considering the torque acting on the CoM of each link giving the final equilibrium equation as

$$\sum_{v=1}^{V} \mathbf{T}_k^v(L_v+1) = -\sum_{v=1}^{V}\sum_{l=1}^{L_v} \tau_k^v(l) - \tau_k^b. \tag{54}$$

The full derivation is given in [16]. As for the forces, $-\mathbf{T}_k^v(L_v+1)$ is the external torque acting on the L_vth link.

3.2 Inverse Dynamics from Motion Capture Data

Assuming the location of the CoM, the principal axes of inertia of each link and the 3D marker positions attached to the links, it is possible to calculate the inertial force and torque quantities relating to the subject of the motion capture. Assuming the frame rate of the capture system is constant with an interval of P, given the bivectors $B_{k+1}^T(l)$, $B_k^T(l)$, $B_{k-1}^T(l)$, the translations $\mathbf{t}_{k+1}(l)$, $\mathbf{t}_k(l)$, $\mathbf{t}_{k-1}(l)$ and the $\mathbf{c}_k(l)$, the results in the previous sections can be summarised as;

$$\begin{aligned}
\dot{B}_k^T(l) &\approx \frac{B_{k+1}^T(l) - B_{k-1}^T(l)}{2P}, \ddot{B}_k^T(l) \approx \frac{B_{k+1}^T(l) - 2B_k^T(l) + B_{k-1}^T(l)}{P^2} \\
\ddot{\mathbf{t}}_k(l) &\approx \frac{\mathbf{t}_{k+1}(l) - 2\mathbf{t}_k(l) + \mathbf{t}_{k-1}(l)}{P^2} \\
\theta_k(l) &= 2\sqrt{-B_k^T(l)B_k^T(l)}, \quad \hat{B}_k^T(l) = \frac{2B_k^T(l)}{\theta_k(l)}
\end{aligned}$$

$$
\dot{\theta}_k(l) = -4\frac{\dot{B}_k^T(l)\underline{\times}B_k^T(l)}{\theta_k(l)}, \quad \dot{\hat{B}}_k^T(l) = \frac{2\dot{B}_k^T(l) - \dot{\theta}_k(l)\hat{B}_k^T(l)}{\theta_k(l)}
$$

$$
\Omega_k^S(l) = \dot{\theta}_k(l)\hat{B}_k^T(l) + 2\sin(\theta_k(l)/2)\dot{\hat{B}}_k^T(l)R_k^{\dagger T}(l)
$$

$$
\ddot{\theta}_k(l) = -\left(\frac{4\left(\ddot{B}_k^T(l)\underline{\times}B_k^T(l)\right) + 4\left(\dot{B}_k^T(l)\right)^2 + \left(\dot{\theta}_k(l)\right)^2}{\theta_k(l)}\right)
$$

$$
\begin{aligned}
\ddot{\hat{B}}_k^T(l) &= \frac{2\ddot{B}_k^T(l) - \ddot{\theta}_k(l)\hat{B}_k^T(l) - 2\dot{\theta}_k(l)\dot{\hat{B}}_k^T(l)}{\theta_k(l)} \\
\dot{\Omega}_k^S(l) &= \ddot{\theta}_k(l)\hat{B}_k^T(l) + \dot{\theta}_k(l)\dot{\hat{B}}_k^T(l) + \dot{\theta}_k(l)\cos(\theta_k(l)/2)\dot{\hat{B}}_k^T(l)R_k^{\dagger T}(l) \\
&\quad + 2\sin(\theta_k(l)/2)\ddot{\hat{B}}_k^T(l)R_k^{\dagger T}(l) + \sin(\theta_k(l)/2)\dot{\hat{B}}_k^T(l)R_k^{\dagger T}(l)\Omega_k^S(l)
\end{aligned}
$$

$$
\begin{aligned}
\ddot{\mathbf{c}}_k(l) &= \mathbf{c}_k(l) \times \dot{\Omega}_k^S(l) + \left[\mathbf{c}_k(l) \times \Omega_k^S(l)\right] \times \Omega_k^S(l) && (55) \\
F_k(l) &= m(l)\left(\ddot{\mathbf{c}}_k(l) + \ddot{\mathbf{t}}_k(l)\right) \\
\Omega_k^B(l) &= R_k^{\dagger T}(l)\Omega_k^S(l)R_k^T(l), \dot{\Omega}_k^B(l) = R_k^{\dagger T}(l)\dot{\Omega}_k^S(l)R_k^T(l) \\
\tau_k(l) &= R_k^T(l)\left[\mathcal{I}(\Omega_k^B(l)) \times \Omega_k^B(l) + \mathcal{I}(\dot{\Omega}_k^B(l))\right] R_k^{\dagger T}(l) && (56)
\end{aligned}
$$

where $\Omega_k^S(l)$ and $\Omega_k^B(l)$ are the 'space' and 'body' angular velocity bivectors respectively.

If the data from the whole system is available it is possible to apply the equilibrium equations (53) and (54) to estimate the external forces and torques acting on the system up to a scale factor using

$$
\sum_{v=1}^{V}\mathbf{F}_k^v(L_v+1) = \sum_{v=1}^{V}\sum_{l=1}^{L_v}m_v(l)\mathbf{g} - \sum_{v=1}^{V}\sum_{l=1}^{L_v}m_v(l)\left[\ddot{\mathbf{c}}_k^v(l) + \ddot{\mathbf{t}}_k^v(l)\right] - m(b)\left[\ddot{\mathbf{c}}_k^b + \ddot{\mathbf{t}}_k^b\right]
$$

and

$$
\sum_{v=1}^{V}\mathbf{T}_k^v(L_v+1) = -\sum_{v=1}^{V}\sum_{l=1}^{L_v}\tau_k(l) - \tau_k^b
$$

where $\ddot{\mathbf{c}}_k^v(l)$ and $\ddot{\mathbf{c}}_k^b$ are evaluated from equation (55) and $\tau_k(l)$ and τ_k^b are evaluated using equation (56). Note that $\ddot{\mathbf{c}}_k^v(l) + \ddot{\mathbf{t}}_k^v(l)$ is the acceleration of the CoM of the lth link at time k of the branch v.

4 Real world applications and results

The above techniques were applied to a dataset obtained from a bridge simulator [20]. In this case the bridge was oscillating with one degree of freedom and the human subject walking on a treadmill on the bridge phase locked into the bridge oscillation. We have assumed that the oscillation direction is horizontal even though in the actual simulator it has a vertical component. Eight markers were placed on the joints of the legs of the human subject and three markers were

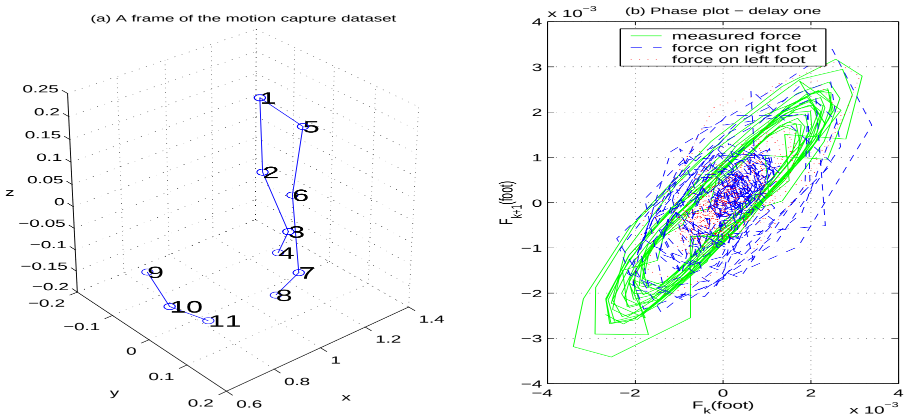


Figure 5: (a) A frame of the motion capture data showing the legs of the subject and the bridge markers. (b) The phase plot of the mechanically measured force on the bridge and the forces calculated from the motion capture data.

placed on the bridge as shown in figure (5(a)). The markers were captured using a motion capture system described in [21]. The output of the system is a set of 3D marker positions. Since in this particular dataset a single marker per limb is placed at the joints, calculation of the joint location and the rotation from one frame to the next was trivial. As this is not accurate, only qualitatively accurate results were expected in this experiment. The limbs were modelled as solid cylinders with axes assumed parallel to the vector between two joint markers. The principal axes and the inertia tensors were calculated accordingly. The rotation from principal axes to the first frame ($r_r(l)$) was calculated using the GA method given in [22]. The rotational bivectors were calculated from the rotors representing the rotations from the principal axes to each frame. The calculated bivectors were smoothed using Savitzky-Golay filters as implemented in [18] since the data is noisy and first and second derivatives of the bivectors wrt time must be evaluated. The procedure described in section (3.2) was then applied to the resulting data. $\mathbf{t}_{k+1}(l)$ and $\mathbf{c}_k(l)$ were trivially calculated in this case as the joint marker location and the half-way point on the corresponding link, respectively.

An example of the results are shown in figure (6) – many more plots are given in [16]. Also the actual displacement of the bridge was measured mechanically. This displacement data was used to calculate the bridge acceleration. The phase relationship of the acceleration of the bridge versus that of the foot is compared in figure (5(b)).

These figures are presented as an illustration of an application of the procedure described in this paper. For example, note also that data from figure (6) suggests that the very marked twist of the foot towards the end of the gait cycle (before the foot is raised) seems to be responsible for most of the lateral force on the bridge from the walker. It is also clear that scalar quantities such as the angular velocity of the foot (and indeed the lower leg and the upper leg although this data is not shown here) make smooth cyclic patterns in all directions. Further work will compare the gait patterns between walkers on swinging and stationary structures. These techniques can be used to complement data from force plate measurements and can also be used directly for biomedical engineering applications.

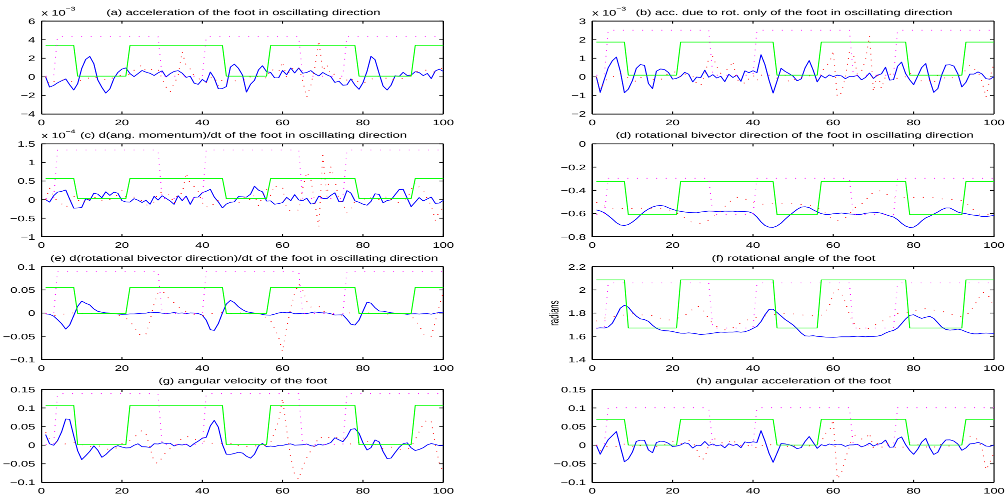


Figure 6: (a)-(e) Inverse dynamics quantities of the foot in the approximate oscillating direction of the bridge plane. (f)-(h) the absolute angle of rotation from the principal axis to the current position, angular velocity and angular acceleration of the foot. The dotted lines are left limb and the solid lines are right limb data in all figures. The rectangular pulses represent the times the corresponding leg is in contact with the bridge. Note that the y-axis is accurate only up to a scale factor unless labelled.

4.1 Conclusions and future work

Here we have described an algorithm to estimate the quantities relevant in inverse dynamics from the 3D positions of the points on moving articulated bodies. Although we have mainly concentrated on data obtained visually, the techniques can be readily applied to other technologies, such as magnetic markers. Most of the methods given here can also be used in robotics. In the application dataset given, the joint locations and the rotations were estimated trivially. But in general, if there are multiple markers per link, these quantities can be calculated in a least squares sense using the techniques described in [23] and [19]. The crucial reason for the resulting simple recipe for calculating inverse dynamic quantities is the use of a single GA quantity, the rotational bivector, as the variable quantity rather than treating direction and magnitudes separately.

Clearly more experimental work is necessary to validate the procedures described here. Ideally the algorithm should be cast in a probabilistic framework and also, a sensitivity analysis, similar to that given in [10], should be carried out. Such an analysis would estimate the sensitivity of our derived quantities to errors in the locations of points, models used etc.

We believe that, despite current limitations, the methods we have described here provide a set of powerful tools for estimation of dynamical quantities for use in engineering, biomedical applications and computer animation.

References

[1] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. D. Reidel: Dordrecht, 1984.

[2] D. Hestenes, *New Foundations for Classical Mechanics*. Kluwer Academic Publishers, 2nd edition., 1999.

[3] C. Doran and A. Lasenby, "Physical applications of geometric algebra," Available at `http://www.mrao.cam.ac.uk/ clifford/ptIIIcourse/`. 1999.

[4] J. Lasenby, W. Fitzgerald, A. Lasenby, and C. Doran, "New geometric methods for computer vision: An application to structure and motion estimation," *International Journal of Computer Vision*, vol. 26, no. 3, pp. 191–213, 1998.

[5] L. Dorst, S. Mann, and T. Bouma, "GABLE: A MATLAB tutorial for geometric algebra," Available at `http://www.carol.wins.uva,nl/ leo/clifford/gablebeta.html`.2000.

[6] H. Goldstein, "Classical Mechanics," Second Edition. Addison Wesley. 1980.

[7] J.M McCarthy, "An Introduction to Theoretical Kinematics". MIT Press, Cambridge, MA. 1990.

[8] H. Li, D. Hestenes, A. Rockwood. " Generalized Homogeneous Coordinates for Computational Geometry". G. Sommer, editor, *Geometric Computing with Clifford Algebras*. Springer. To appear Summer 2001.

[9] D. Hestenes, D. "Old wine in new bottles: a new algebraic framework for computational geometry". Advances in Geometric Algebra with Applications in Science and Engineering, Eds. Bayro-Corrochano, E.D. and Sobcyzk, G. Birkhauser Boston. To appear Summer 2000.

[10] B. Karan and M. Vukobratović, "Calibration and accuracy of manipulation robot models - an overview," *Mechanism and Machine Theory*, vol. 29, no. 3, pp. 479–500, 1994.

[11] H. Asada and J. J. E. Slotine, *Robot Analysis and Control*. John Wiley and Sons, 1986.

[12] J. Luh, W. M.W., and P. R.P.C, "On-line computational scheme for mechanical manipulators," *A.S.M.E. J. Dyn. Syst. Meas. Contr.*, no. 102, 1980.

[13] J. Apkarian, S. Naumann, and B. Cairns, "A three-dimentional kinematic and dynamic model of the lower limb," *Journal of Biomechanics*, vol. 22, no. 2, pp. 143–155, 1989.

[14] C. Doran and A. Lasenby, "Physical applications of geometric algebra," 2001. Available at `http://www.mrao.cam.ac.uk/˜clifford/ptIIIcourse/`.

[15] D. Hestenes, *New Foundations for Classical Mechanics*. Kluwer Academic Publishers, second ed., 1999.

[16] S. S. Hiniduma Udugama Gamage and J. Lasenby, "Extraction of rigid body dynamics from motion capture data," CUED Technical Report CUED/F-INFENG/TR.410, Cambridge University Engineering Department, 2001.

[17] F. A. McRobie and J. Lasenby, "Simo-Vu Quoc rods using Clifford algebra," *International Journal for Numerical Methods in Engineering*, vol. 45, pp. 377–398, 1999.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, second ed., 1992.

[19] S. S. Hiniduma Udugama Gamage, M. Ringer, and J. Lasenby, "Estimation of centres and axes of rotation of articulated bodies in general motion for global skeleton fitting," CUED Technical Report CUED/F-INFENG/TR.408, Cambridge University Engineering Department, 2001.

[20] A. McRobie, G. Morgenthal, J. Lasenby, M. Ringer, and S. Gamage, "Millennium bridge simulator." Available at http://www2.eng.cam.ac.uk/˜gm249/MillenniumBridge/.

[21] M. Ringer and J. Lasenby, "Modelling and tracking articulated motion from muliple camera views," in *Analysis of Biomedical Signals and Images: 15-th Biennial International EURASIP conference Euroconference Biosignal 2000 Proceedings* (J. Jan, J. Kozumpík, I. Provazník, and Z. Szabó, eds.), pp. 92–94, Brno University of Technology Vutium Press, 2000.

[22] J. Lasenby, W. Fitzgerald, A. Lasenby, and C. Doran, "New geometric methods for computer vision: An application to structure and motion estimation," *International Journal of Computer Vision*, vol. 26, no. 3, pp. 191–213, 1998.

[23] S. S. Hiniduma Udugama Gamage and J. Lasenby, "A new least squares solution for estimation of centre and axis of rotation," CUED Technical Report CUED/F-INFENG/TR.399, Cambridge University Engineering Department, 2001.

Geometric Algebra: New Foundations, New Insights

Three-Dimensional Geometric Algebra and Rotations

Dr. Chris Doran
Cavendish Laboratory
Madingley Road
Cambridge University, CB3 0HE, UK

c.doran@mrao.cam.ac.uk

Dr. Joan Lasenby
Department of Engineering
University of Cambridge
Trumpington Street
Cambridge, CB2 1PZ, UK

jl@eng.cam.ac.uk

www.mrao.cam.ac.uk/~clifford

1 Geometric Algebra of 3-D Space

The geometric algebra (GA) of 3-d space is a remarkably powerful tool for solving problems in geometry and classical mechanics. It describes vectors, planes and volumes in a single algebra, which contains all of the familiar vector operations for 3-d space. These include the vector cross product, which is revealed as a disguised form of bivector. The algebra provides a very clear and compact method for encoding rotations, which is considerably more powerful than working with matrices. This reveals the true significance of Hamilton's quaternions, and resolves many of the historical difficulties encountered with their use.

As a basis set for the geometric algebra of 3-d space we use chose a set of orthonormal vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$. All three vectors are perpendicular, so they all *anticommute*, and have unit length so they square to $+1$. From these 3 basis vectors we can generate 3 independent bivectors:

$$\boldsymbol{e}_1\boldsymbol{e}_2, \quad \boldsymbol{e}_2\boldsymbol{e}_3, \quad \text{and} \quad \boldsymbol{e}_3\boldsymbol{e}_1. \tag{1.1}$$

Each of these encodes a distinct plane, and there are 3 of them to match the 3 independent planes in 3-d space. As well as the 3 bivectors the algebra contains one further object. This is the product of 3 orthogonal vectors, resulting in

$$(\boldsymbol{e}_1\boldsymbol{e}_2)\boldsymbol{e}_3 = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3. \tag{1.2}$$

This corresponds to sweeping the bivector $\boldsymbol{e}_1\boldsymbol{e}_2$ along the vector $\boldsymbol{e}_3$. The result is a 3-dimensional volume element and is called a *trivector*. This is said to have *grade*-3, where the word 'grade' refers to the number of independent vectors forming the object. So vectors are grade-1, bivectors are grade-2, and so on. The term 'grade' is preferred to 'dimension' as the latter is reserved for the size of a linear space.

In 3-d the maximum number of independent vectors is 3, so the trivector is the highest grade object, or *multivector*, in the algebra. This trivector is unique up to scale (*i.e.* volume) and handedness (see below). The unit highest-grade multivector is called the *pseudoscalar*, or *directed volume element*. The latter name is more accurate, but the former is seen more often. (Though be careful with this usage — pseudoscalar can mean different things in different contexts). To simplify, we introduce the symbol I,

$$I = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3. \tag{1.3}$$

Our 3-d algebra is therefore spanned by

$$\begin{array}{cccc} 1 & \{\boldsymbol{e}_i\} & \{\boldsymbol{e}_i \wedge \boldsymbol{e}_j\} & I = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3 \\ \text{1 scalar} & \text{3 vectors} & \text{3 bivectors} & \text{1 trivector} \end{array} \tag{1.4}$$

These define a linear space of dimension $8 = 2^3$. We call this algebra $\mathcal{G}_3$. Notice that the dimensions of each subspace are given by the binomial coefficients.

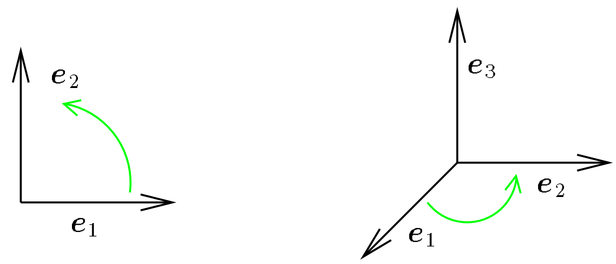


Figure 1: *Handedness.* The two frames shown are, by convention, assigned a right-handed orientation. Both $\boldsymbol{e}_1\boldsymbol{e}_2$ and $\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3$ give rise to right-handed pseudoscalars for their respective algebras.

The pseudoscalar is, by convention, chosen to be *right-handed.* This is equivalent to saying that the generating frame $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ is right-handed. If a left-handed set of orthonormal vectors is multiplied together the result is $-I$. There is no intrinsic definition of handedness — it is a convention adopted to make our life easier. In 3-d a right-handed frame is constructed as follows. Align your thumb along the $\boldsymbol{e}_3$ direction. Then the grip of your right hand specifies the direction in which $\boldsymbol{e}_1$ rotates onto $\boldsymbol{e}_2$ (Fig. 1). The handedness of a frame changes sign if the positions of any two vectors are swapped.

2 Products in $\mathcal{G}_3$

Any two vectors in the algebra, $\boldsymbol{a}$ and $\boldsymbol{b}$ say, can be multiplied with the geometric product, and we have

$$\boldsymbol{ab} = \boldsymbol{a}\cdot\boldsymbol{b} + \boldsymbol{a}\wedge\boldsymbol{b}. \tag{2.1}$$

Now the bivector $\boldsymbol{a}\wedge\boldsymbol{b}$ belongs to a 3-d space, spanned by the $\{\boldsymbol{e}_i\wedge\boldsymbol{e}_j\}$. If we expand out in a basis,

$$\boldsymbol{a} = \sum_{i=1}^{3} a_i\boldsymbol{e}_i, \quad \boldsymbol{b} = \sum_{i=1}^{3} b_i\boldsymbol{e}_i, \tag{2.2}$$

we find that the components of the outer product are given by

$$\boldsymbol{a}\wedge\boldsymbol{b} = (a_2b_3 - b_3a_2)\boldsymbol{e}_2\wedge\boldsymbol{e}_3 + (a_3b_1 - a_1b_3)\boldsymbol{e}_3\wedge\boldsymbol{e}_1 + (a_1b_2 - a_2b_1)\boldsymbol{e}_1\wedge\boldsymbol{e}_2. \tag{2.3}$$

The components are the same as those of the cross product, but the result is a bivector rather than a vector. To understand the relationship between these we first need to establish the properties of some of the new products provided by our 3-d algebra.

2.1 Vectors and Bivectors

The three basis bivectors satisfy

$$(\boldsymbol{e}_1\boldsymbol{e}_2)^2 = (\boldsymbol{e}_2\boldsymbol{e}_3)^2 = (\boldsymbol{e}_3\boldsymbol{e}_1)^2 = -1 \tag{2.4}$$

and each bivector generates $90^{\rm O}$ rotations in its own plane. So, for example, we see that

$$\boldsymbol{e}_1(\boldsymbol{e}_1\wedge\boldsymbol{e}_2) = \boldsymbol{e}_1(\boldsymbol{e}_1\boldsymbol{e}_2) = \boldsymbol{e}_2, \tag{2.5}$$

which returns a vector. The geometric product for vectors extends to all objects in the algebra, so we can form expressions such as $\boldsymbol{a}B$, where B is a general bivector. But we have now seen that $\boldsymbol{e}_1(\boldsymbol{e}_2\wedge\boldsymbol{e}_3)$ is a trivector, so the result of the product $\boldsymbol{a}B$ can clearly contain both vector and trivector terms. To help understand the properties of the product $\boldsymbol{a}B$ we first decompose $\boldsymbol{a}$ into terms in and out of the plane,

$$\boldsymbol{a} = \boldsymbol{a}_{||} + \boldsymbol{a}_{\perp} \tag{2.6}$$

(see Fig. 2). We can now write $\boldsymbol{a}B = (\boldsymbol{a}_{||} + \boldsymbol{a}_{\perp})B$. Suppose that we also write

$$B = \boldsymbol{a}_{||}\wedge\boldsymbol{b} \tag{2.7}$$

where $\boldsymbol{b}$ is orthogonal to $\boldsymbol{a}_{||}$ in the B plane (Fig. 2). We see that

$$\boldsymbol{a}_{||}B = \boldsymbol{a}_{||}(\boldsymbol{a}_{||}\wedge\boldsymbol{b}) = \boldsymbol{a}_{||}(\boldsymbol{a}_{||}\boldsymbol{b}) = (\boldsymbol{a}_{||})^2\boldsymbol{b} \tag{2.8}$$

which is a vector in the $\boldsymbol{b}$ direction. On the other hand

$$\boldsymbol{a}_{\perp}B = \boldsymbol{a}_{\perp}(\boldsymbol{a}_{||}\wedge\boldsymbol{b}) = \boldsymbol{a}_{\perp}\boldsymbol{a}_{||}\boldsymbol{b} \tag{2.9}$$

is the geometric product of 3 orthogonal vectors, and so is a trivector. As expected, the geometric product of the vector $\boldsymbol{a}$ and the bivector B has resulted in two terms, a vector and a trivector. We therefore write

$$\boldsymbol{a}B = \boldsymbol{a}\cdot B + \boldsymbol{a}\wedge B \tag{2.10}$$

where the dot is generalised to mean the *lowest* grade part of the result, while the wedge means the *highest* grade part of the result.

2.2 Inner Product $\boldsymbol{a}\cdot B$

From Eq. (2.8) we see that the $\boldsymbol{a}\cdot B = \boldsymbol{a}_{||}\cdot B$ term projects onto the component of $\boldsymbol{a}$ in the plane, and then rotates this through $90^{\rm O}$ and dilates by the magnitude of B. We also see that

$$\boldsymbol{a}\cdot B = {\boldsymbol{a}_{||}}^2\boldsymbol{b} = -(\boldsymbol{a}_{||}\boldsymbol{b})\boldsymbol{a}_{||} = -B\cdot\boldsymbol{a}, \tag{2.11}$$

$$\boldsymbol{a} = \boldsymbol{a}_{\|} + \boldsymbol{a}_{\perp}$$

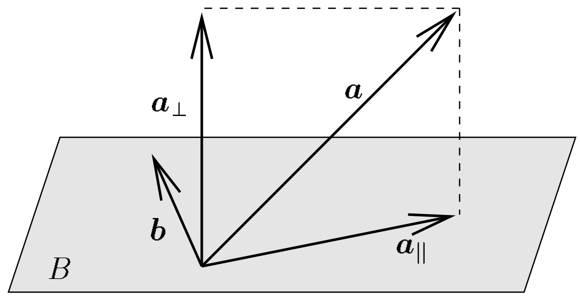


Figure 2: *A vector and a plane.* The vector $\boldsymbol{a}$ is decomposed into a sum of two vectors, one lying in the plane and the other perpendicular to it.

so the dot product between a vector and a bivector is antisymmetric. We use this to *define* the inner product of a vector and a bivector as

$$\boldsymbol{a}\cdot B = \tfrac{1}{2}(\boldsymbol{a}B - B\boldsymbol{a}). \tag{2.12}$$

To see that this always returns a vector, consider the inner product $\boldsymbol{a}\cdot(\boldsymbol{b}\wedge\boldsymbol{c})$. Following the rules for the geometric product we form:

$$\begin{aligned}\boldsymbol{a}(\boldsymbol{b}\wedge\boldsymbol{c}) =& \tfrac{1}{2}\boldsymbol{a}(\boldsymbol{bc} - \boldsymbol{cb}) \\ =& (\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{c} - (\boldsymbol{a}\cdot\boldsymbol{c})\boldsymbol{b} - \tfrac{1}{2}(\boldsymbol{bac} - \boldsymbol{cab}) \\ =& 2(\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{c} - 2(\boldsymbol{a}\cdot\boldsymbol{c})\boldsymbol{b} + \tfrac{1}{2}(\boldsymbol{bc} - \boldsymbol{cb})\boldsymbol{a} \\ =& 2(\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{c} - 2(\boldsymbol{a}\cdot\boldsymbol{c})\boldsymbol{b} + (\boldsymbol{b}\wedge\boldsymbol{c})\boldsymbol{a},\end{aligned} \tag{2.13}$$

where we have made repeated use of the rearrangement

$$\boldsymbol{ba} = 2\boldsymbol{a}\cdot\boldsymbol{b} - \boldsymbol{ab}. \tag{2.14}$$

It follows immediately that

$$\boldsymbol{a}\cdot(\boldsymbol{b}\wedge\boldsymbol{c}) = \tfrac{1}{2}\big(\boldsymbol{a}(\boldsymbol{b}\wedge\boldsymbol{c}) - (\boldsymbol{b}\wedge\boldsymbol{c})\boldsymbol{a}\big) = (\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{c} - (\boldsymbol{a}\cdot\boldsymbol{c})\boldsymbol{b}, \tag{2.15}$$

which is indeed a pure vector. This is one of the most useful results in geometric algebra and is worth memorising.

2.3 Outer Product $\boldsymbol{a}\wedge B$

From Eq. (2.9), the $\boldsymbol{a}\wedge B$ term projects onto the component perpendicular to the plane, and returns a trivector. This term is symmetric

$$\boldsymbol{a}\wedge B = \boldsymbol{a}_{\perp}\boldsymbol{a}_{\|}\boldsymbol{b} = \boldsymbol{a}_{\|}\boldsymbol{b}\boldsymbol{a}_{\perp} = B\wedge\boldsymbol{a}. \tag{2.16}$$

We therefore define the outer product of a vector and a bivector as

$$\boldsymbol{a}\wedge B = \tfrac{1}{2}(\boldsymbol{a}B + B\boldsymbol{a}). \tag{2.17}$$

Various arguments can be used to show that this is a pure trivector (see later). We now have a definition of the outer product of three vectors, $\boldsymbol{a}\wedge(\boldsymbol{b}\wedge\boldsymbol{c})$. This is the grade-3 part of the geometric product. We denote the operation of projecting onto the terms of a given grade with the $\langle\,\rangle_r$ symbol, where r is the required grade. Using this we can write

$$\boldsymbol{a}\wedge(\boldsymbol{b}\wedge\boldsymbol{c}) = \langle \boldsymbol{a}(\boldsymbol{b}\wedge\boldsymbol{c})\rangle_3 = \langle \boldsymbol{a}(\boldsymbol{b}\boldsymbol{c} - \boldsymbol{b}\cdot\boldsymbol{c})\rangle_3. \tag{2.18}$$

But in the final term $\boldsymbol{a}(\boldsymbol{b}\cdot\boldsymbol{c})$ is a vector (grade-1) so does not contribute. It follows that

$$\boldsymbol{a}\wedge(\boldsymbol{b}\wedge\boldsymbol{c}) = \langle \boldsymbol{a}(\boldsymbol{b}\boldsymbol{c})\rangle_3 = \langle \boldsymbol{a}\boldsymbol{b}\boldsymbol{c}\rangle_3, \tag{2.19}$$

where we have used the fact that the geometric product is associative to remove the brackets. It follows from this simple derivation that the outer product is also associative,

$$(\boldsymbol{a}\wedge\boldsymbol{b})\wedge\boldsymbol{c} = \boldsymbol{a}\wedge(\boldsymbol{b}\wedge\boldsymbol{c}) = \boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c}. \tag{2.20}$$

This is true in general.

The trivector $\boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c}$ can be pictured as the parallelepiped formed by sweeping $\boldsymbol{a}\wedge\boldsymbol{b}$ along $\boldsymbol{c}$ (see Fig. 3). The same result is obtained by sweeping $\boldsymbol{b}\wedge\boldsymbol{c}$ along $\boldsymbol{a}$, which is the geometric way of picturing the associativity of the outer product. The other main property of the outer product is that it is antisymmetric on every pair of vectors,

$$\boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c} = -\boldsymbol{b}\wedge\boldsymbol{a}\wedge\boldsymbol{c} = \boldsymbol{c}\wedge\boldsymbol{a}\wedge\boldsymbol{b}, \quad \textit{etc.} \tag{2.21}$$

This expresses the geometric result that swapping any two vectors reverses the orientation (handedness) of the product.

2.4 The Bivector Algebra

Our three independent bivectors also give us a further new product to consider. When multiplying two bivectors we find, for example, that

$$(\boldsymbol{e}_1\wedge\boldsymbol{e}_2)(\boldsymbol{e}_2\wedge\boldsymbol{e}_3) = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_2\boldsymbol{e}_3 = \boldsymbol{e}_1\boldsymbol{e}_3, \tag{2.22}$$

resulting in a third bivector. We also find that

$$(\boldsymbol{e}_2\wedge\boldsymbol{e}_3)(\boldsymbol{e}_1\wedge\boldsymbol{e}_2) = \boldsymbol{e}_3\boldsymbol{e}_2\boldsymbol{e}_2\boldsymbol{e}_1 = \boldsymbol{e}_3\boldsymbol{e}_1 = -\boldsymbol{e}_1\boldsymbol{e}_3, \tag{2.23}$$

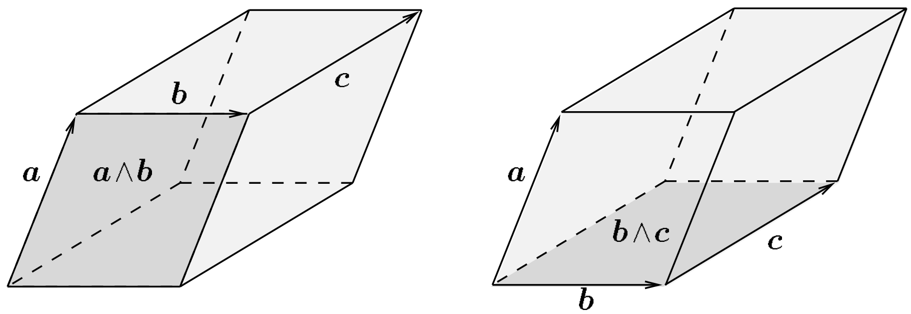


Figure 3: *The Trivector.* The result of sweeping $\boldsymbol{a}\wedge\boldsymbol{b}$ along $\boldsymbol{c}$ is a directed volume, or trivector. The same trivector is obtained by sweeping $\boldsymbol{b}\wedge\boldsymbol{c}$ along $\boldsymbol{a}$.

so the product is antisymmetric. The symmetric contribution vanishes because the two planes are perpendicular. If we introduce the following labelling for the basis bivectors:

$$B_1 = \boldsymbol{e}_2\boldsymbol{e}_3, \quad B_2 = \boldsymbol{e}_3\boldsymbol{e}_1, \quad B_3 = \boldsymbol{e}_1\boldsymbol{e}_2 \tag{2.24}$$

we find that the commutator satisfies

$$B_iB_j - B_jB_i = -2\epsilon_{ijk}B_k. \tag{2.25}$$

This algebra is closely linked to 3-d rotations, and will be familiar from the quantum theory of angular momentum.

It is useful to introduce a symbol for one-half the commutator of 2 bivectors. We call this the *commutator product* and denote it with a cross, so

$$A\times B = \tfrac{1}{2}(AB - BA). \tag{2.26}$$

The commutator product of two bivectors always results in a third bivector (or zero).

The basis bivectors all square to -1, and all anticommute. These are the properties of the generators of the quaternion algebra. This observation helps to sort out some of the problems encountered with the quaternions. Hamilton attempted to identify pure quaternions (null scalar part) with vectors, but we now see that they are actually *bivectors*. This has an important consequence when we look at their behaviour under reflections. Hamilton also imposed the condition $\boldsymbol{ijk} = -1$ on his unit quaternions, whereas we have

$$B_1B_2B_3 = \boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_3\boldsymbol{e}_1\boldsymbol{e}_1\boldsymbol{e}_2 = +1. \tag{2.27}$$

To set up a direct map we must flip a sign somewhere, for example in the y component:

$$\boldsymbol{i} \leftrightarrow B_1, \quad \boldsymbol{j} \leftrightarrow -B_2, \quad \boldsymbol{k} \leftrightarrow B_3. \tag{2.28}$$

This shows us that the quaternions were *left-handed*, even though the $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ were interpreted as a right-handed set of vectors. Not surprisingly, this was a source of some confusion!

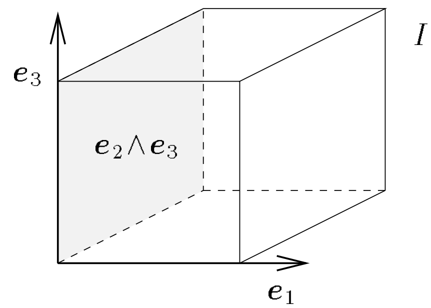


Figure 4: *The product of a vector and a trivector.* The diagram shows the result of the product $\boldsymbol{e}_1 I = \boldsymbol{e}_1(\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3) = \boldsymbol{e}_2\boldsymbol{e}_3$

2.5 Products Involving the Pseudoscalar

The pseudoscalar $I = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3$ is the unique right-handed unit trivector in the algebra. This gives us a number of new products to consider. We start by forming the product of I with the vector $\boldsymbol{e}_1$,

$$I\boldsymbol{e}_1 = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_1 = -\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_1\boldsymbol{e}_3 = \boldsymbol{e}_2\boldsymbol{e}_3. \tag{2.29}$$

The result is a bivector — the plane perpendicular to the original vector (see Fig. 4). The product of a grade-1 vector with the grade-3 pseudoscalar is therefore a grade-2 bivector. Reversing the order we find that

$$\boldsymbol{e}_1 I = \boldsymbol{e}_1\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3 = \boldsymbol{e}_2\boldsymbol{e}_3. \tag{2.30}$$

The result is therefore independent of order — the pseudoscalar commutes with all vectors in 3-d,

$$I\boldsymbol{a} = \boldsymbol{a}I, \quad \text{for all } \boldsymbol{a}. \tag{2.31}$$

It follows that I commutes with all elements in the algebra. This is always the case for the pseudoscalar in spaces of odd dimension. In even dimensions, the pseudoscalar anticommutes with all vectors, as can be easily checked in 2-d. We can now express each of our basis bivectors as the product of the pseudoscalar and a *dual* vector,

$$\boldsymbol{e}_1\boldsymbol{e}_2 = I\boldsymbol{e}_3, \qquad \boldsymbol{e}_2\boldsymbol{e}_3 = I\boldsymbol{e}_1, \qquad \boldsymbol{e}_3\boldsymbol{e}_1 = I\boldsymbol{e}_2. \tag{2.32}$$

This operation of multiplying by the pseudoscalar is called a *duality* transformation.

We next form the square of the pseudoscalar

$$I^2 = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3 = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_1\boldsymbol{e}_2 = -1. \tag{2.33}$$

So the pseudoscalar commutes with all elements and squares to -1. It is therefore a further candidate for a unit imaginary. In some physical applications this is the correct one to use, whereas for others it is one of the bivectors. These different possibilities provide us with a very rich geometric language.

Finally, we consider the product of a bivector and the pseudoscalar:

$$I(\boldsymbol{e}_1\wedge\boldsymbol{e}_2) = I\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_3 = II\boldsymbol{e}_3 = -\boldsymbol{e}_3. \tag{2.34}$$

So the result of the product of I with the bivector formed from $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ is $-\boldsymbol{e}_3$, that is, minus the vector perpendicular to the $\boldsymbol{e}_1\wedge\boldsymbol{e}_2$ plane. This affords a definition of the vector cross product in 3-d as

$$\boldsymbol{a}\boldsymbol{\times}\boldsymbol{b} = -I(\boldsymbol{a}\wedge\boldsymbol{b}). \tag{2.35}$$

The bold $\boldsymbol{\times}$ symbol should not be confused with the $\times$ symbol for the commutator product. The latter is extremely useful, whereas the vector cross product is largely redundant now that we have the outer product available. Equation (2.35) shows how the cross product is a bivector in disguise, the bivector being mapped to a vector by a duality operation. It is also now clear why the product only exists in 3-d — this is the only space for which the dual of a bivector is a vector. We will have little further use for the cross product and will rarely employ it from now on. This means we can also do away with the awkward distinction between axial and polar vectors. Instead we just talk of vectors and bivectors.

The duality operation in 3-d provides an alternative way to understand the geometric product $\boldsymbol{a}B$ of a vector and a bivector. We write $B = I\boldsymbol{b}$ in terms of its dual vector $\boldsymbol{b}$, so that we now have

$$\boldsymbol{a}B = I\boldsymbol{a}\boldsymbol{b} = I(\boldsymbol{a}\cdot\boldsymbol{b} + \boldsymbol{a}\wedge\boldsymbol{b}). \tag{2.36}$$

This demonstrates that the symmetric part of the product generates the trivector

$$\boldsymbol{a}\wedge B = I(\boldsymbol{a}\cdot\boldsymbol{b}) = \tfrac{1}{2}(\boldsymbol{a}B + B\boldsymbol{a}), \tag{2.37}$$

whereas the antisymmetric part returns a vector

$$\boldsymbol{a}\cdot B = I(\boldsymbol{a}\wedge\boldsymbol{b}) = \tfrac{1}{2}(\boldsymbol{a}B - B\boldsymbol{a}). \tag{2.38}$$

This justifies the definition of the inner and outer products between a vector and bivector. As with pairs of vectors, these combine to return the geometric product,

$$\boldsymbol{a}B = \boldsymbol{a}\cdot B + \boldsymbol{a}\wedge B. \tag{2.39}$$

3 Further Definitions

An important operation in GA is that of reversing the order of vectors in any product. This is denoted with a dagger, $A^{\dagger}$. Scalars and vectors are invariant under reversion, but bivectors change sign,

$$(\boldsymbol{e}_1\boldsymbol{e}_2)^\dagger = \boldsymbol{e}_2\boldsymbol{e}_1 = -\boldsymbol{e}_1\boldsymbol{e}_2. \tag{3.1}$$

Similarly, we see that

$$I^\dagger = \boldsymbol{e}_3\boldsymbol{e}_2\boldsymbol{e}_1 = \boldsymbol{e}_1\boldsymbol{e}_3\boldsymbol{e}_2 = -\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3 = -I. \tag{3.2}$$

A general multivector in 3-d can be written

$$M = \alpha + \boldsymbol{a} + B + \beta I. \tag{3.3}$$

From the above we see that

$$M^\dagger = \alpha + \boldsymbol{a} - B - \beta I. \tag{3.4}$$

The choice of the dagger symbol reflects the fact that, if one chooses to adopt a Hermitian matrix representation for the vector generators, the reverse operation corresponds to the Hermitian adjoint for matrices.

It is also useful to adopt the *operator ordering convention* that, in the absence of brackets, *inner and outer products are performed before geometric products.* This cleans up expressions by enabling us to remove unnecessary brackets. For example, on the right-hand side of Eq. (2.35) we can now write

$$\boldsymbol{a}\times\boldsymbol{b} = -I\,\boldsymbol{a}\wedge\boldsymbol{b}. \tag{3.5}$$

We have already introduced the $\langle\,\rangle_r$ notation for projecting onto the terms of grade-r. For the operation of projecting onto the scalar component we usually drop the subscript 0 and write

$$\langle AB\rangle = \langle AB\rangle_0 \tag{3.6}$$

for the scalar part of the product of two arbitrary multivectors. The scalar product is always symmetric

$$\langle AB\rangle = \langle BA\rangle. \tag{3.7}$$

It follows that

$$\langle A\cdots BC\rangle = \langle CA\cdots B\rangle. \tag{3.8}$$

This cyclic reordering property is very useful in practice.

$$\begin{aligned} \boldsymbol{a} &= \boldsymbol{a}_\perp + \boldsymbol{a}_{||} \\ \boldsymbol{a}' &= \boldsymbol{a}_\perp - \boldsymbol{a}_{||} \end{aligned}$$

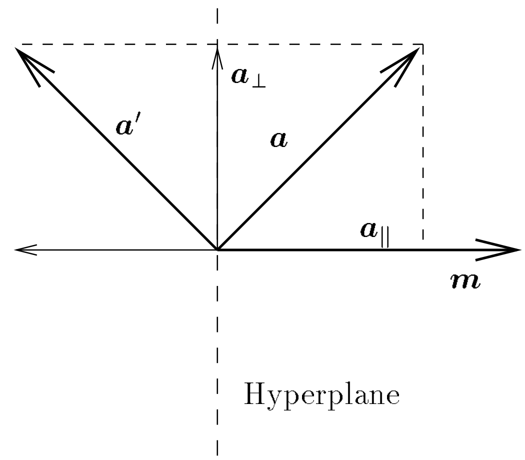


Figure 5: *A reflection in the plane perpendicular to* $\boldsymbol{m}$.

4 Reflections

Suppose that we reflect the vector $\boldsymbol{a}$ in the (hyper)plane orthogonal to some unit vector $\boldsymbol{m}$ ($\boldsymbol{m}^2 = 1$). The component of $\boldsymbol{a}$ parallel to $\boldsymbol{m}$ changes sign, whereas the perpendicular component is unchanged. The parallel component is the projection onto $\boldsymbol{m}$:

$$\boldsymbol{a}_{||} = \boldsymbol{a}\cdot\boldsymbol{m}\,\boldsymbol{m}. \tag{4.1}$$

(NB operator ordering convention in force here.) The perpendicular component is the remainder

$$\boldsymbol{a}_\perp = \boldsymbol{a} - \boldsymbol{a}\cdot\boldsymbol{m}\,\boldsymbol{m} = (\boldsymbol{a}\boldsymbol{m} - \boldsymbol{a}\cdot\boldsymbol{m})\boldsymbol{m} = \boldsymbol{a}\wedge\boldsymbol{m}\,\boldsymbol{m}. \tag{4.2}$$

This shows how the wedge product projects onto the components perpendicular to a vector. The result of the reflection is therefore

$$\begin{aligned} \boldsymbol{a}' = \boldsymbol{a}_\perp - \boldsymbol{a}_{||} &= -\boldsymbol{a}\cdot\boldsymbol{m}\,\boldsymbol{m} + \boldsymbol{a}\wedge\boldsymbol{m}\,\boldsymbol{m} \\ &= -(\boldsymbol{m}\cdot\boldsymbol{a} + \boldsymbol{m}\wedge\boldsymbol{a})\boldsymbol{m} = -\boldsymbol{m}\boldsymbol{a}\boldsymbol{m}. \end{aligned} \tag{4.3}$$

This remarkably compact formula only arises in geometric algebra. We can start to see now that geometric products arise naturally when *operating* on vectors.

It is simple to check that our formula has the required properties. For any vector $\lambda\boldsymbol{m}$ in the $\boldsymbol{m}$ direction we have

$$-\boldsymbol{m}(\lambda\boldsymbol{m})\boldsymbol{m} = -\lambda\boldsymbol{m}\boldsymbol{m}\boldsymbol{m} = -\lambda\boldsymbol{m} \tag{4.4}$$

and so $\lambda\boldsymbol{m}$ is reflected. Similarly, for any vector $\boldsymbol{n}$ perpendicular to $\boldsymbol{m}$ we have

$$-\boldsymbol{m}(\boldsymbol{n})\boldsymbol{m} = -\boldsymbol{mnm} = \boldsymbol{nmm} = \boldsymbol{n} \tag{4.5}$$

and so $\boldsymbol{n}$ is unaffected. We can also give a simple proof that inner products are unchanged by reflections,

$$\begin{aligned} \boldsymbol{a}'\cdot\boldsymbol{b}' &= (-\boldsymbol{mam})\cdot(-\boldsymbol{mbm}) = \langle\boldsymbol{mammbm}\rangle \\ &= \langle\boldsymbol{mabm}\rangle = \langle\boldsymbol{mmab}\rangle = \boldsymbol{a}\cdot\boldsymbol{b}. \end{aligned} \tag{4.6}$$

We next construct the transformation law for the bivector $\boldsymbol{a}\wedge\boldsymbol{b}$ under reflection of both $\boldsymbol{a}$ and $\boldsymbol{b}$. We obtain

$$\begin{aligned} \boldsymbol{a}'\wedge\boldsymbol{b}' &= (-\boldsymbol{mam})\wedge(-\boldsymbol{mbm}) = \tfrac{1}{2}(\boldsymbol{mammbm} - \boldsymbol{mbmmam}) \\ &= \tfrac{1}{2}\boldsymbol{m}(\boldsymbol{ab} - \boldsymbol{ba})\boldsymbol{m} = \boldsymbol{m}\,\boldsymbol{a}\wedge\boldsymbol{b}\,\boldsymbol{m}. \end{aligned} \tag{4.7}$$

We recover essentially the same law, but with a crucial sign difference. Bivectors do not transform as vectors under reflections. This is the reason for the confusing distinction between polar and axial vectors in 3-d. Axial vectors invariably arise as the result of the cross product. They are really bivectors and should be treated as such. This also explains why 19th century mathematicians were confused by the transformation properties of the quaternions. They were expected to transform as vectors under reflections, but actually transform as bivectors (*i.e.* with the opposite sign).

5 Rotations

For many years, Hamilton struggled with the problem of finding a compact representation for rotations in 3-d. His goal was to generalise to representation of 2-d rotations as a complex phase change. The key to finding the correct formula is to use that result that *a rotation in the plane generated by two unit vectors* $\boldsymbol{m}$ *and* $\boldsymbol{n}$ *is achieved by successive reflections in the (hyper)planes perpendicular to* $\boldsymbol{m}$ *and* $\boldsymbol{n}$*.* This is illustrated in Fig. 6. It is clear that any component of $\boldsymbol{a}$ outside the $\boldsymbol{m}\wedge\boldsymbol{n}$ plane is untouched. It is also a simple exercise in trigonometry to confirm that the angle between the initial vector $\boldsymbol{a}$ and the final vector $\boldsymbol{a}''$ is twice the angle between $\boldsymbol{m}$ and $\boldsymbol{n}$. The result of the successive reflections is therefore to rotate through 2θ in the $\boldsymbol{m}\wedge\boldsymbol{n}$ plane, where $\boldsymbol{m}\cdot\boldsymbol{n} = \cos(\theta)$.

So how does this look in GA?

$$\boldsymbol{a}' = -\boldsymbol{mam} \tag{5.1}$$

$$\boldsymbol{a}'' = -\boldsymbol{na}'\boldsymbol{n} = -\boldsymbol{n}(-\boldsymbol{mam})\boldsymbol{n} = \boldsymbol{nmamn} \tag{5.2}$$

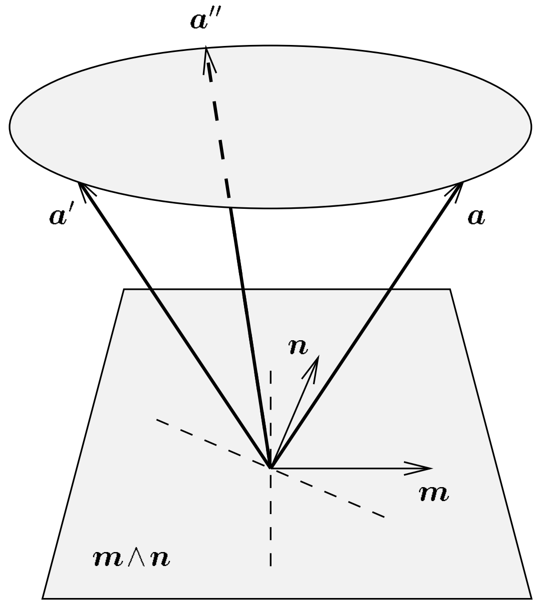


Figure 6: *A Rotation from 2 Reflections.* $\boldsymbol{a}'$ is the result of reflecting $\boldsymbol{a}$ in the plane perpendicular to $\boldsymbol{m}$. $\boldsymbol{a}''$ is the result of reflecting $\boldsymbol{a}'$ in the plane perpendicular to $\boldsymbol{n}$.

This is beginning to look very simple! We define the *rotor* R by

$$R = \boldsymbol{n}\boldsymbol{m}. \tag{5.3}$$

Note the *geometric* product here! We can now write a rotation as

$$\boldsymbol{a} \mapsto R\boldsymbol{a}R^{\dagger} \tag{5.4}$$

Incredibly, this formula works for any grade of multivector, in any dimension, of any signature! To make contact with the 2-d result we first expand R as

$$R = \boldsymbol{n}\boldsymbol{m} = \boldsymbol{n}\cdot\boldsymbol{m} + \boldsymbol{n}\wedge\boldsymbol{m} = \cos(\theta) + \boldsymbol{n}\wedge\boldsymbol{m}. \tag{5.5}$$

So what is the magnitude of the bivector $\boldsymbol{n}\wedge\boldsymbol{m}$?

$$\begin{aligned}(\boldsymbol{n}\wedge\boldsymbol{m})\cdot(\boldsymbol{n}\wedge\boldsymbol{m}) &= \langle \boldsymbol{n}\wedge\boldsymbol{m}\,\boldsymbol{n}\wedge\boldsymbol{m}\rangle = \langle \boldsymbol{n}\boldsymbol{m}\,\boldsymbol{n}\wedge\boldsymbol{m}\rangle \\ &= \boldsymbol{n}\cdot[\boldsymbol{m}\cdot(\boldsymbol{n}\wedge\boldsymbol{m})] = \boldsymbol{n}\cdot(\boldsymbol{m}\cos(\theta) - \boldsymbol{n}) \\ &= \cos^2(\theta) - 1 = -\sin^2(\theta).\end{aligned} \tag{5.6}$$

We therefore define a unit bivector in the $\boldsymbol{m}\wedge\boldsymbol{n}$ plane by

$$\hat{B} = \boldsymbol{m}\wedge\boldsymbol{n}/\sin(\theta), \qquad \hat{B}^2 = -1. \tag{5.7}$$

This choice of orientation ($\boldsymbol{m}\wedge\boldsymbol{n}$ rather than $\boldsymbol{n}\wedge\boldsymbol{m}$) ensures that the bivector has the same orientation as the rotation, as can be seen in Fig. 6.

In terms of the bivector $\hat{B}$ we now have

$$R = \cos(\theta) - \hat{B}\sin(\theta). \tag{5.8}$$

Look familiar? This is nothing else than the polar decomposition of a complex number, with the unit imaginary replaced by the unit bivector $\hat{B}$. We can therefore write

$$R = \exp(-\hat{B}\theta). \tag{5.9}$$

The exponential here is defined in terms of its power series in the normal way. It is possible to show that this series is absolutely convergent for any multivector argument. (Exponentiating a multivector is essentially the same as exponentiating a matrix).

Now recall that our formula was for a rotation through 2θ. If we want to rotate through θ, the appropriate rotor is

$$R = \exp\{-\hat{B}\theta/2\} \tag{5.10}$$

which gives us the final formula

$$\boldsymbol{a} \mapsto \mathrm{e}^{-\hat{B}\theta/2}\,\boldsymbol{a}\,\mathrm{e}^{\hat{B}\theta/2}. \tag{5.11}$$

This describes a rotation through θ in the $\hat{B}$ plane, with orientation specified by $\hat{B}$. The GA description forces us to think of rotations taking place *in a plane* as opposed to about an axis. The latter is an entirely 3-d concept, whereas the concept of a plane is quite general.

Rotors are one of the fundamental concepts in geometric algebra. Since the rotor R is a geometric product of two unit vectors, we see immediately that

$$RR^\dagger = \boldsymbol{nm}(\boldsymbol{nm})^\dagger = \boldsymbol{nmmn} = 1 = R^\dagger R. \tag{5.12}$$

This provides a quick proof that our formula has the correct property of preserving lengths and angles,

$$\boldsymbol{a}'\cdot\boldsymbol{b}' = (R\boldsymbol{a}R^\dagger)\cdot(R\boldsymbol{b}R^\dagger) = \langle R\boldsymbol{a}R^\dagger R\boldsymbol{b}R^\dagger\rangle = \langle R\boldsymbol{ab}R^\dagger\rangle = \boldsymbol{a}\cdot\boldsymbol{b}. \tag{5.13}$$

Now suppose that the two vectors forming the bivector $B = \boldsymbol{a}\wedge\boldsymbol{b}$ are both rotated. What is the expression for the resulting bivector? To find this we form

$$\begin{aligned} B' = \boldsymbol{a}'\wedge\boldsymbol{b}' &= \tfrac{1}{2}(\boldsymbol{a}'\boldsymbol{b}' - \boldsymbol{b}'\boldsymbol{a}') = \tfrac{1}{2}(R\boldsymbol{a}R^\dagger R\boldsymbol{b}R^\dagger - R\boldsymbol{b}R^\dagger R\boldsymbol{a}R^\dagger) \\ &= \tfrac{1}{2}(R\boldsymbol{ab}R^\dagger - R\boldsymbol{ba}R^\dagger) = \tfrac{1}{2}R(\boldsymbol{ab} - \boldsymbol{ba})R^\dagger = R\boldsymbol{a}\wedge\boldsymbol{b}R^\dagger = RBR^\dagger. \end{aligned} \tag{5.14}$$

Bivectors are rotated using precisely the same formula as vectors! The same turns out to be true for all geometric objects represented by multivectors. This is one of the most attractive features of geometric algebra.

6 Properties of Rotors

Let us consider the problem of rotating a unit vector $\boldsymbol{n}_1$ into another unit vector $\boldsymbol{n}_2$ in 3-d space, where the angle between these two vectors in θ. What is the rotor R which performs such a rotation? If R is the rotor we require then it must satisfy $\boldsymbol{n}_2 = R\boldsymbol{n}_1 R^\dagger$ which, under multiplication on the right by R gives,

$$\boldsymbol{n}_2 R = R\boldsymbol{n}_1. \tag{6.1}$$

Now consider the quantity $(1 + \boldsymbol{n}_2\boldsymbol{n}_1)$. Since $\boldsymbol{n}_1{}^2 = \boldsymbol{n}_2{}^2 = 1$, we see that

$$\boldsymbol{n}_2(1 + \boldsymbol{n}_2\boldsymbol{n}_1) = \boldsymbol{n}_2 + \boldsymbol{n}_1 \tag{6.2}$$

$$(1 + \boldsymbol{n}_2\boldsymbol{n}_1)\boldsymbol{n}_1 = \boldsymbol{n}_1 + \boldsymbol{n}_2 \tag{6.3}$$

so equation (6.1) is satisfied if $R \propto (1 + \boldsymbol{n}_2\boldsymbol{n}_1)$. It remains simply to normalize R so that it satisfies $RR^\dagger = 1$. If $R = \alpha(1 + \boldsymbol{n}_2\boldsymbol{n}_1)$ we obtain

$$RR^\dagger = \alpha^2(1 + \boldsymbol{n}_2\boldsymbol{n}_1)(1 + \boldsymbol{n}_1\boldsymbol{n}_2) = 2\alpha^2(1 + \boldsymbol{n}_2\cdot\boldsymbol{n}_1), \tag{6.4}$$

which gives us the following formula for R:

$$R = \frac{1 + \boldsymbol{n}_2\boldsymbol{n}_1}{\sqrt{2(1 + \boldsymbol{n}_2\cdot\boldsymbol{n}_1)}}. \tag{6.5}$$

We can recover our earlier expression by first noting that

$$\sqrt{2(1 + \boldsymbol{n}_2\cdot\boldsymbol{n}_1)} = 2\cos(\theta/2). \tag{6.6}$$

The rotor R can be now written as

$$R = \cos(\theta/2) + \frac{\boldsymbol{n}_2\wedge\boldsymbol{n}_1}{|\boldsymbol{n}_2\wedge\boldsymbol{n}_1|}\sin(\theta/2) = \exp\left(-\frac{\theta}{2}\frac{\boldsymbol{n}_1\wedge\boldsymbol{n}_2}{|\boldsymbol{n}_2\wedge\boldsymbol{n}_1|}\right), \tag{6.7}$$

where $|\boldsymbol{n}_2 \wedge \boldsymbol{n}_1|$ is the magnitude of the bivector $\boldsymbol{n}_2 \wedge \boldsymbol{n}_1$, defined by

$$|\boldsymbol{n}_2 \wedge \boldsymbol{n}_1| = \Big((\boldsymbol{n}_2\wedge\boldsymbol{n}_1)\cdot(\boldsymbol{n}_1\wedge\boldsymbol{n}_2)\Big)^{1/2}. \tag{6.8}$$

In this way the rotor is again written as the exponential of a bivector, recovering Eq. (5.9). An alternative representation, available only in 3-d, is to introduce the *dual* vector $\boldsymbol{n}$ and write the rotor as

$$R = \exp\left(-I\frac{\theta}{2}\boldsymbol{n}\right) = \cos(\theta/2) - I\boldsymbol{n}\sin(\theta/2). \tag{6.9}$$

This generates a rotation of θ radians about an axis parallel to the unit vector n in a right-handed screw sense. (This is precisely how 3-D rotations are represented in the quaternion algebra.)

6.1 Composition Law

A feature of the rotor treatment of rotations is the ease with which rotations can now be combined. Suppose that the rotor R_1 takes the vector $\boldsymbol{a}$ to the vector $\boldsymbol{b}$,

$$\boldsymbol{b} = R_1 \boldsymbol{a} R_1^\dagger. \tag{6.10}$$

If the vector $\boldsymbol{b}$ is now rotated by a second rotor R_2 to the vector $\boldsymbol{c}$, we have

$$\boldsymbol{c} = R_2 \boldsymbol{b} R_2^\dagger, \tag{6.11}$$

and therefore

$$\boldsymbol{c} = (R_2 R_1)\boldsymbol{a}(R_2 R_1)^\dagger. \tag{6.12}$$

The combined rotation is therefore generated by the composite rotor

$$R = R_2 R_1. \tag{6.13}$$

This is the *group composition* law for rotors. It is straightforward to check that this results in a new rotor.

This composition rule has two important features. The first is that in finding the composite rotor a *maximum* of 16 multiplications is required. This compares favourably with the 27 required when multiplying together 2 rotation matrices. The second is that we have far better control over numerical errors when combining rotors. If numerical errors do arise, the worst that can happen is that the rotor is no longer normalised exactly to 1. This is easily rectified by rescaling. No such simple method is available with rotation matrices. If numerical errors mean that the matrix is no longer orthogonal there is no simple method to recover the "nearest" orthogonal matrix.

6.2 Frames and Reciprocals

A frequently-encountered problem is how to find the rotor given two arbitrary sets of vectors, known to be related my a rotation. To solve this problem we must first introduce the notion of a *reciprocal* frame. Given a set of linearly independent vectors $\{\boldsymbol{e}_i\}$ (where now no assumption of orthonormality is made), the reciprocal frame, $\{\boldsymbol{e}^i\}$, is defined such that

$$\boldsymbol{e}^i \cdot \boldsymbol{e}_j = \delta^i_j. \tag{6.14}$$

We construct such a reciprocal frame in n-dimensions as follows:

$$\boldsymbol{e}^j = (-1)^{j-1} \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \cdots \wedge \check{\boldsymbol{e}}_j \wedge \cdots \wedge \boldsymbol{e}_n \, I_e^{-1} \tag{6.15}$$

where $I_e = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \cdots \wedge \boldsymbol{e}_n$ and $\check{\boldsymbol{e}}_j$ indicates that $\boldsymbol{e}_j$ is missing from the product. In three dimensions this is a very simple operation and the reciprocal frame vectors for a linearly independent set of vectors $\{\boldsymbol{e}_i\}$ are as follows:

$$\begin{aligned}
\boldsymbol{e}^1 &= \frac{1}{\alpha} I \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \\
\boldsymbol{e}^2 &= \frac{1}{\alpha} I \boldsymbol{e}_3 \wedge \boldsymbol{e}_1 \\
\boldsymbol{e}^3 &= \frac{1}{\alpha} I \boldsymbol{e}_1 \wedge \boldsymbol{e}_2,
\end{aligned} \tag{6.16}$$

where $I\alpha = \boldsymbol{e}_3 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_1$.

A vector $\boldsymbol{a}$ can be expanded in either frame as follows (summation convention in force)

$$\boldsymbol{a} = a_j \boldsymbol{e}^j \equiv (\boldsymbol{a} \cdot \boldsymbol{e}_j) \boldsymbol{e}^j \tag{6.17}$$

$$\boldsymbol{a} = a^j \boldsymbol{e}_j \equiv (\boldsymbol{a} \cdot \boldsymbol{e}^j) \boldsymbol{e}_j. \tag{6.18}$$

The identification of a_j with $\boldsymbol{a} \cdot \boldsymbol{e}_j$ is obtained by dotting the equation $\boldsymbol{a} = a_j \boldsymbol{e}^j$ with $\boldsymbol{e}_i$. Similarly, the identification of a^j with $\boldsymbol{a} \cdot \boldsymbol{e}^j$ is obtained by dotting $\boldsymbol{a} = a^j \boldsymbol{e}_j$ with $\boldsymbol{e}^i$. So, given a general frame and a vector, the reciprocal frame is needed to construct the components of the vector in the chosen frame. Of course, for orthonormal frames there is no distinction between the frame and its reciprocal.

Suppose now that we have two sets of vectors in 3-d (not necessarily orthonormal) $\{\boldsymbol{e}_k\}$ and $\{\boldsymbol{f}_k\}$ which we know are related by a rotation. We hence know that

$$\boldsymbol{f}_k = R \boldsymbol{e}_k R^\dagger \tag{6.19}$$

and we seek a simple expression for the rotor R. As we are in 3-d, we can write

$$R = \mathrm{e}^{-B/2} \quad \text{and} \quad R^\dagger = \mathrm{e}^{B/2} = \cos(|B|/2) + \sin(|B|/2) B/|B|. \tag{6.20}$$

We therefore find that

$$\begin{aligned}
\boldsymbol{e}_k R^\dagger \boldsymbol{e}^k &= \boldsymbol{e}_k [\cos(|B|/2) + \sin(|B|/2) B/|B|] \boldsymbol{e}^k \\
&= 3\cos(|B|/2) - \sin(|B|/2) B/|B| \\
&= 4\cos(|B|/2) - R^\dagger.
\end{aligned} \tag{6.21}$$

We now form

$$\boldsymbol{f}_k \boldsymbol{e}^k = R \boldsymbol{e}_k R^\dagger \boldsymbol{e}^k = 4\cos(|B|/2) R - 1. \tag{6.22}$$

It follows that R is a scalar multiple of $1 + \boldsymbol{f}_k \boldsymbol{e}^k$. We therefore establish the simple formula

$$R = \frac{1 + \boldsymbol{f}_k \boldsymbol{e}^k}{|1 + \boldsymbol{f}_k \boldsymbol{e}^k|} = \frac{\psi}{\sqrt{(\psi \psi^\dagger)}} \tag{6.23}$$

where $\psi = 1 + \boldsymbol{f}_k \boldsymbol{e}^k$. This neat formula recovers the rotor directly from the frame vectors. It works in all cases except when the rotation is through 180^{o}, in which case $\psi = 0$. This is easily handled as a special case.

6.3 Rotation Matrices

The conventional way to treat rotations is through the application of 3×3 orthogonal matrices, which are applied to the coordinates of a vector in a given fixed orthonormal frame. If we denote this frame by $\{\boldsymbol{e}_k\}$ we have $\boldsymbol{a} = a_k \boldsymbol{e}_k$ and

$$\boldsymbol{a}' = R\boldsymbol{a}R^\dagger = a'_k \boldsymbol{e}_k. \tag{6.24}$$

The components of the rotated vector $\boldsymbol{a}'$ are related to the original components by

$$a'_i = \mathsf{R}_{ij} a_j \tag{6.25}$$

where R is an orthogonal matrix. From the preceding we have

$$a'_i = \boldsymbol{e}_i \cdot \boldsymbol{a}' = \boldsymbol{e}_i \cdot (R\boldsymbol{a}R^\dagger) = \boldsymbol{e}_i \cdot (R\boldsymbol{e}_j R^\dagger) a_j. \tag{6.26}$$

It follows that the matrix components are given by

$$\mathsf{R}_{ij} = \boldsymbol{e}_i \cdot (R\boldsymbol{e}_j R^\dagger). \tag{6.27}$$

As expected, the components depend quadratically on the rotor R. It follows that R and $-R$ encode *the same* rotation. Even for the simplest rotations, one can see that the rotor encoding is significantly more compact than the matrix expression.

Given a rotation matrix R_{ij} one can recover the rotor efficiently by adapting Eq. (6.23). We define

$$\psi = 1 + \mathsf{R}_{ij} \boldsymbol{e}_i \boldsymbol{e}_j = 1 + R\boldsymbol{e}_j R^\dagger \boldsymbol{e}_j \tag{6.28}$$

so that the rotor is given by

$$R = \frac{\psi}{\sqrt{(\psi\psi^\dagger)}}. \tag{6.29}$$

This result makes it easy to convert from the standard formulation to the geometric algebra framework.

6.4 Euler Angles

The Standard *Euler angle* formulation of rotations is to express any rotation as a combination of 3 rotations:

1st: rotate ϕ about the $\boldsymbol{e}_3$ axis
2nd: rotate θ about the **rotated** $\boldsymbol{e}_1$ axis
3rd: rotate ψ about the **rotated** $\boldsymbol{e}_3$ axis

In traditional accounts this is involves defining a set of 3 rotation matrices

$$\mathsf{A}_1 = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathsf{A}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}$$

$$\mathsf{A}_3 = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.30}$$

we are then told to apply these matrices in *reverse* order to form

$$\mathsf{A} = \mathsf{A}_1\mathsf{A}_2\mathsf{A}_3 \tag{6.31}$$

so that the coordinates transform as $x' = \mathsf{A}x$. This matrix ordering is often confusing and is justified using arguments based on mixtures of "active" and "passive" transformations. It is therefore instructive to see how this looks in geometric algebra.

We start by defining the rotor

$$R_1 = \exp\left(-I\frac{\phi}{2}\boldsymbol{e}_3\right), \tag{6.32}$$

which generates a rotation about the $\boldsymbol{e}_3$ axis. Next we need a rotation about the rotated $\boldsymbol{e}_1$ axis, which is generated by

$$R_2 = \exp\left(-I\frac{\theta}{2}\boldsymbol{e}_1'\right) \tag{6.33}$$

where $\boldsymbol{e}_1' = R_1\boldsymbol{e}_1R_1^\dagger$. One can see, then, that

$$R_2 = R_1\exp\left(-I\frac{\theta}{2}\boldsymbol{e}_1\right)R_1^\dagger. \tag{6.34}$$

Finally, we rotate about the *new* 3-axis, which requires the rotor

$$R_3 = \exp\left(-I\frac{\psi}{2}\boldsymbol{e}_3''\right) \tag{6.35}$$

where $\boldsymbol{e}_3'' = R_2R_1\boldsymbol{e}_3R_1^\dagger R_2^\dagger$. In this case the rotor can be written as

$$R_3 = R_2R_1\exp\left(-I\frac{\psi}{2}\boldsymbol{e}_3\right)R_1^\dagger R_2^\dagger. \tag{6.36}$$

Now forming the combined rotor R we find that

$$\begin{aligned} R = R_3R_2R_1 &= R_2R_1\exp\left(-I\frac{\psi}{2}\boldsymbol{e}_3\right)R_1^\dagger R_2^\dagger R_2R_1 \\ &= R_1\exp\left(-I\frac{\theta}{2}\boldsymbol{e}_1\right)R_1^\dagger R_1\exp\left(-I\frac{\psi}{2}\boldsymbol{e}_3\right) \\ &= \exp\left(-I\frac{\phi}{2}\boldsymbol{e}_3\right)\exp\left(-I\frac{\theta}{2}\boldsymbol{e}_1\right)\exp\left(-I\frac{\psi}{2}\boldsymbol{e}_3\right). \end{aligned} \tag{6.37}$$

This fully explains the order in which the rotations are applied, and avoids all complications connected with changing frames midway through the calculation, or attempting to distinguish rotations of coordinates, rotations of coordinate axes, and ("active") rotations of vectors.

Despite the clean form of the Euler angle formalism in geometric algebra, this is rarely an optimal encoding for rotations. Given an arbitrary rotor, its decomposition into Euler angles is not straightforward, and the product formula is equally messy. In practice it is best to either work directly with the rotor R, or with its bivector generator B, $R = \exp(-B/2)$.

6.5 Interpolating Rotors

Rotors are elements of a four-dimensional space, normalised to 1. They can be represented as points on a *3-sphere* — the set of unit vectors in four dimensions. This is the rotor *group manifold*. At any point on the manifold, the *tangent space* is three-dimensional. This is the analog of the tangent plane to a sphere in three dimensions. Rotors therefore require three parameters to specify them uniquely. The simplest choice of parameters is directly in terms of the bivector generators, with

$$|B^2| \le \pi. \tag{6.38}$$

The rotors R and $-R$ generate the *same* rotation, because of their double-sided action. It follows that the *rotation* group manifold is more complicated than the rotor group manifold — it is a projective 3-sphere with points R and $-R$ identified. This is one reason why it is usually easier to work with rotors.

Suppose we are given two estimates of a rotation, R_0 and R_1, how do we find the mid-point? With rotors this is remarkably easy! Suppose that the rotors are R_0 and R_1. We first make sure they have the smallest angle between them in four dimensions. This is done by ensuring that

$$\langle R_0 R_1^\dagger \rangle = \cos\theta > 0. \tag{6.39}$$

If this inequality is not satisfied, then the sign of one of the rotors should be flipped. The 'shortest' path between the rotors on the group manifold is defined by

$$R(\lambda) = R_0 \exp(\lambda B), \tag{6.40}$$

where

$$R(0) = R_0, \quad R(1) = R_1. \tag{6.41}$$

It follows that we can find B from

$$\exp(B) = R_0^\dagger R_1. \tag{6.42}$$

The path defined by $\exp(\lambda B)$ is an invariant construct. If both endpoints are transformed, the path transforms in the same way. The midpoint is

$$R_{1/2} = R_0 \exp(B/2), \tag{6.43}$$

which therefore generates the midpoint rotation. This is quite general — it works for any rotor group (or any *Lie group*). For rotations in three dimensions we can do even better. R_0 and R_1 can be viewed as two unit vectors in a four-dimensional space. The path $\exp(\lambda B)$ lies in the plane specified by these vectors: the rotors can therefore be treated as unit vectors in four dimensions. The path between them lies entirely in the plane of the two rotors, and therefore defines a segment of a circle.

The rotor path between R_0 and R_1 can be written as

$$R(\lambda) = R_0\big(\cos\lambda\theta\, + \sin\lambda\theta\,\hat{B}\big), \tag{6.44}$$

where we have used $B = \theta\hat{B}$. But we know that

$$\exp(B) = \cos\theta\, + \sin\theta\,\hat{B} = R_0^\dagger R_1. \tag{6.45}$$

It follows that

$$\begin{aligned} R(\lambda) &= \frac{R_0}{\sin\theta}\big(\sin\theta\,\cos\lambda\theta\, + \sin\lambda\theta(R_0^\dagger R_1 - \cos\theta)\big) && (6.46)\\ &= \frac{1}{\sin\theta}\big(\sin(1-\lambda)\theta\, R_0 + \sin\lambda\theta\, R_1\big), && (6.47) \end{aligned}$$

which satisfies $R(\lambda)R^\dagger(\lambda) = 1$ for all λ. The midpoint rotor is therefore simply

$$R_{1/2} = \frac{\sin(\theta/2)}{\sin\theta}(R_0 + R_1). \tag{6.48}$$

This gives us a remarkably simple prescription for finding the midpoint: *add the rotors and normalise the result.* By comparison, the equivalent matrix is quadratic in R, and so is much more difficult to express in terms of the two endpoint rotation matrices.

Suppose now that we have a number of estimates for a rotation and wanted to find the average. Again the answer is simple. First one chooses the sign of the rotors so that they are all in the 'closest' configuration. This will normally be easy if the rotations are all roughly equal. If some of the rotations are quite different then one might have to search around for the closest configuration, though in these cases the average of such rotations is not a useful concept. Once one has all of the rotors chosen, one simply adds them up and normalises the result to obtain the average. This sort of calculation can be useful in computer vision problems where one has a number of estimates of the relative rotations between cameras, and their average is required.

The lesson here is that problems involving rotations can be simplified by working with rotors and relaxing the normalisation criteria. This enables us to work in a four-dimensional linear space and is the basis for a simplified calculus for rotations.

7 Differentiation for multivector quantities

Here we give a brief discussion of the process of differentiating with respect to any multivector. Having available such a calculus means that, in practice, it is easy to take derivatives with respect to rotors and vectors and this makes many least-squares minimization problems much simpler to deal with. In computer vision and motion analysis one tends to draw frequently on an approach which minimizes some expression in order to find the relevant rotations and translations – this is a standard technique for any estimation problem in the presence of uncertainty.

If X is a mixed-grade multivector, $X = \sum_r X_r$, and $F(X)$ is a general multivector-valued function of X, then the derivative of F in the A 'direction' is written as $A * \partial_X F(X)$ (here we use $*$ to denote the scalar part of the product of two multivectors, i.e. $A * B \equiv \langle AB \rangle$), and is defined as

$$A * \partial_X F(X) \equiv \lim_{\tau \to 0} \frac{F(X + \tau A) - F(X)}{\tau}. \tag{7.1}$$

For the limit on the right hand side to make sense A must contain only grades which are contained in X and no others. If X contains no terms of grade-r and A_r is a homogeneous multivector, then we define $A_r * \partial_X = 0$. This definition of the derivative also ensures that the operator $A * \partial_X$ is a scalar operator and satisfies all of the usual partial derivative properties. We can now use the above definition of the directional derivative to formulate a general expression for the multivector derivative ∂_X without reference to one particular direction. This is accomplished by introducing an arbitrary frame $\{\boldsymbol{e}_j\}$ and extending this to a basis (vectors, bivectors, etc..) for the entire algebra, $\{\boldsymbol{e}_J\}$. Then ∂_X is defined as

$$\partial_X \equiv \sum_J \boldsymbol{e}^J (\boldsymbol{e}_J * \partial_X), \tag{7.2}$$

where $\{\boldsymbol{e}^J\}$ is an extended basis built out of the reciprocal frame. The directional derivative, $\boldsymbol{e}_J * \partial_X$, is only non-zero when $\boldsymbol{e}_J$ is one of the grades contained in X (as previously discussed) so that ∂_X inherits the multivector properties of its argument X. Although we have here defined the multivector derivative using an extended basis, it should be noted that the sum over all the basis ensures that ∂_X is independent of the choice of $\{\boldsymbol{e}_j\}$ and so all of the properties of ∂_X can be formulated in a frame-free manner. One of the most useful results concerning multivector derivatives is

$$\partial_X \langle XB \rangle = B, \tag{7.3}$$

where we assume that B and X contain the same grades. (If the grades are different then only the terms in B which share grades with X are produced on the right.) From this basic result one can also see that

$$\partial_X \langle X^\dagger B \rangle = \partial_X \langle X B^\dagger \rangle = B^\dagger. \tag{7.4}$$

7.1 Rotor Calculus

Any function of a rotation can be viewed as taking its values over the group manifold. In most of what follows we are interested in scalar functions, though there is no reason to restrict to this case. The derivative of the function with respect to a rotor defines a vector in the tangent space at each point on the group manifold. The vector points in the direction of steepest increase of the function. This can all be made mathematically rigorous and is the subject of *differential geometry*. The problem is that much off this is over-complicated for the relatively simple minimisation problems encountered in computer vision. Working intrinsically on the group manifold involves introducing local coordinates (such as the Euler angles) and differentiating with respect to each of these in turn. The resulting calculations can be long and messy and often hide the simplicity of the answer.

Geometric algebra provides us with a more elegant and simpler alternative. We relax the rotor normalisation constraint and replace R by ψ — a general element of the even subalgebra. It is straightforward to show that the derivative operator defined above reduces to a simple form if we first decompose ψ in terms of the $\{\boldsymbol{e}_i\}$ basis as

$$\psi = \psi_0 + \sum_{k=1}^{3} \psi_k I \boldsymbol{e}_k \tag{7.5}$$

where the $\{\psi_0, \dots, \psi_3\}$ are a set of scalar components. The multivector derivative then becomes

$$\partial_\psi = \frac{\partial}{\partial \psi_0} - \sum_{k=1}^{3} I \boldsymbol{e}_k \frac{\partial}{\partial \psi_k}. \tag{7.6}$$

This derivative is independent of the chosen frame. It satisfies the basic result

$$\partial_\psi \langle \psi A \rangle = A \tag{7.7}$$

where A is an even-grade multivector (independent of ψ). All further results for ∂_ψ are built up from this basic result and Leibniz' rule for the derivative of a product.

The basic trick now is to re-write a rotation as

$$R \boldsymbol{a} R^\dagger = \psi \boldsymbol{a} \psi^{-1}. \tag{7.8}$$

This works because any even multivector ψ can be written as

$$\psi = \rho^{1/2} R \tag{7.9}$$

where R is a rotor, $\rho = \psi \psi^\dagger$ and $\rho = 0$ if and only if $\psi = 0$. The inverse of ψ is then

$$\psi^{-1} = \rho^{-1/2} R^\dagger \tag{7.10}$$

so that

$$\psi\psi^{-1} = RR^{\dagger} = 1. \tag{7.11}$$

The equality of equation (7.8) follows immediately. If one imagines a function over a sphere in three dimensions, one can extend this to a function over all space by attaching the same value to all points on each line from the origin. The extension $R \mapsto \psi$ does precisely this, but in a four dimensional space.

We are now able to differentiate functions of the rotation quite simply. The typical application is to a scalar of the type

$$(R\boldsymbol{a}R^{\dagger}) \cdot \boldsymbol{b} = \langle R\boldsymbol{a}R^{\dagger}\boldsymbol{b}\rangle = \langle \psi\boldsymbol{a}\psi^{-1}\boldsymbol{b}\rangle. \tag{7.12}$$

To differentiate this we need a result for the derivative of the inverse of a multivector. We start by letting M be a constant multivector, and derive

$$0 = \partial_{\psi}\langle \psi\psi^{-1}M\rangle = \psi^{-1}M + \dot{\partial}_{\psi}\langle \psi\dot{\psi}^{-1}M\rangle, \tag{7.13}$$

where the overdots denote the scope of the derivative. It follows that

$$\dot{\partial}_{\psi}\langle \dot{\psi}^{-1}M\psi\rangle = -\psi^{-1}M. \tag{7.14}$$

But in this formula we can now let M become a function of ψ, as only the first term, ψ^{-1}, is acted on by the differential operator. We can therefore replace M by $M\psi^{-1}$ to obtain the useful formula

$$\dot{\partial}_{\psi}\langle \dot{\psi}^{-1}M\rangle = -\psi^{-1}M\psi^{-1}. \tag{7.15}$$

So, let us now consider the problem of finding the rotor R which 'most closely' rotates the vectors $\{\boldsymbol{u}_i\}$ onto the vectors $\{\boldsymbol{v}_i\}$, $i = 1, ..., n$. More precisely, we wish to find the rotor R which minimizes

$$\phi = \sum_{i=1}^{n}(\boldsymbol{v}_i - R\boldsymbol{u}_iR^{\dagger})^2. \tag{7.16}$$

Expanding ϕ gives

$$\begin{aligned}\phi &= \sum_{i=1}^{n}(\boldsymbol{v}_i{}^2 - \boldsymbol{v}_iR\boldsymbol{u}_iR^{\dagger} - R\boldsymbol{u}_iR^{\dagger}\boldsymbol{v}_i + R(\boldsymbol{u}_i{}^2)R^{\dagger}) \\ &= \sum_{i=1}^{n}\Big((\boldsymbol{v}_i{}^2 + \boldsymbol{u}_i{}^2) - 2\langle \boldsymbol{v}_iR\boldsymbol{u}_iR^{\dagger}\rangle\Big). \end{aligned} \tag{7.17}$$

We now replace $R\boldsymbol{u}_i R^\dagger$ with $\psi \boldsymbol{u}_i \psi^{-1}$ and differentiate, forming

$$\begin{aligned}\partial_\psi \phi(\psi) &= -2\sum_{i=1}^{n} \partial_\psi \langle \boldsymbol{v}_i \psi \boldsymbol{u}_i \psi^{-1}\rangle \\ &= -2\sum_{i=1}^{n}\Big(\dot{\partial}_\psi \langle \dot{\psi} A_i\rangle + \dot{\partial}_\psi \langle B_i \dot{\psi}^{-1}\rangle\Big),\end{aligned}$$

where $A_i = \boldsymbol{u}_i \psi^{-1} \boldsymbol{v}_i$ and $B_i = \boldsymbol{v}_i \psi \boldsymbol{u}_i$ (using the cyclic reordering property). The first term is easily evaluated to give A_i. To evaluate the second term we can use equation (7.15). One can then substitute $\psi = R$ and note that $R^{-1} = R^\dagger$ as $RR^\dagger = 1$. We then have

$$\begin{aligned}\partial_\psi \phi(\psi) &= -2\sum_{i=1}^{n}\Big(\boldsymbol{u}_i \psi^{-1} \boldsymbol{v}_i - \psi^{-1}(\boldsymbol{v}_i \psi \boldsymbol{u}_i)\psi^{-1}\Big) \\ &= -2\psi^{-1}\sum_{i=1}^{n}\Big((\psi \boldsymbol{u}_i \psi^{-1})\boldsymbol{v}_i - \boldsymbol{v}_i(\psi \boldsymbol{u}_i \psi^{-1})\Big) \\ &= 4R^\dagger \sum_{i=1}^{n} \boldsymbol{v}_i \wedge (R\boldsymbol{u}_i R^\dagger). \qquad (7.18)\end{aligned}$$

Thus the rotor which minimizes the least-squares expression $\phi(R) = \sum_{i=1}^{n}(\boldsymbol{v}_i - R\boldsymbol{u}_i R^\dagger)^2$ must satisfy

$$\sum_{i=1}^{n} \boldsymbol{v}_i \wedge (R\boldsymbol{u}_i R^\dagger) = 0. \qquad (7.19)$$

This is intuitively obvious – we want the R which makes $\boldsymbol{u}_i$ 'most parallel' to $\boldsymbol{v}_i$ in the average sense. The big advantage of the approach used here is that one never leaves the geometric algebra of space, and the resultant bivector is evaluated in the same space, rather than in some abstract tangent space on the group manifold. The solution of equation (7.19) for R will utilize the linear algebra framework of geometric algebra and is described in the following section.

8 Linear algebra

Geometric algebra is a very natural framework for the study of linear functions and non-orthonormal frames. Here we will give a brief account of how geometric algebra deals with linear algebra; we do this since many computer vision and engineering problems can be formulated as problems in linear algebra.

If we take a linear function $\mathsf{F}(\boldsymbol{a})$ which maps vectors to vectors in the same space then it is possible to extend F to act linearly on multivectors. This extension of F is given by

$$\mathsf{F}(\boldsymbol{a}_1\wedge\boldsymbol{a}_2\wedge\ldots\wedge\boldsymbol{a}_r) = \mathsf{F}(\boldsymbol{a}_1)\wedge\mathsf{F}(\boldsymbol{a}_2)\wedge\ldots\wedge\mathsf{F}(\boldsymbol{a}_r). \tag{8.1}$$

The extended function preserves grade since F maps an r-grade multivector to another r-grade multivector. The *adjoint* to F is written as $\bar{\mathsf{F}}$ and defined by

$$\bar{\mathsf{F}}(\boldsymbol{a}) = \boldsymbol{e}^i\langle\mathsf{F}(\boldsymbol{e}_i)\boldsymbol{a}\rangle, \tag{8.2}$$

where, as before, $\{\boldsymbol{e}_i\}$ is an arbitrary frame and $\{\boldsymbol{e}^i\}$ is its reciprocal frame. This definition ensures that

$$\boldsymbol{a}\cdot\mathsf{F}(\boldsymbol{b}) = \boldsymbol{b}\cdot\bar{\mathsf{F}}(\boldsymbol{a}), \tag{8.3}$$

so the adjoint represents the function corresponding to the transpose of the matrix which is represented by F. If $\mathsf{F} = \bar{\mathsf{F}}$ the function is said to be *self-adjoint*, or *symmetric*. Symmetric functions satisfy

$$\boldsymbol{e}_i\wedge\mathsf{F}(\boldsymbol{e}^i) = 0, \tag{8.4}$$

and this ensures that the any matrix representing F is symmetric. Similarly, if $\mathsf{F} = -\bar{\mathsf{F}}$ then the function is antisymmetric.

As an illustration of the use of linear algebra techniques, we will discuss the solution of equation (7.19). We first re-write the equation as

$$\boldsymbol{e}_i\wedge\left[\sum_{i=1}^{n} R\big((\boldsymbol{e}^i\cdot\boldsymbol{v}_i)\boldsymbol{u}_i\big)R^\dagger\right] = 0. \tag{8.5}$$

We now introduce a function F defined by

$$\mathsf{F}(\boldsymbol{a}) = \sum_{i=1}^{n}(\boldsymbol{a}\cdot\boldsymbol{v}_i)\boldsymbol{u}_i. \tag{8.6}$$

Equation (8.5) can then be written as

$$\boldsymbol{e}_i\wedge R\mathsf{F}(\boldsymbol{e}^i)R^\dagger = 0. \tag{8.7}$$

Let us now define another function R mapping vectors onto vectors such that $\mathsf{R}(\boldsymbol{a}) = R\boldsymbol{a}R^\dagger$. With these definitions equation (8.7) takes the form

$$\boldsymbol{e}_i\wedge\mathsf{R}\mathsf{F}(\boldsymbol{e}^i) = 0, \tag{8.8}$$

which tells us that RF is symmetric. We now perform a singular-value decomposition (SVD) on F, which enables us to write

$$\mathsf{F} = \mathsf{SD} \tag{8.9}$$

where S is an orthogonal transformation and D is symmetric. Comparing with (8.8) we see that a solution is provided by

$$\mathsf{R} = \mathsf{S}^{-1} = \bar{\mathsf{S}}. \tag{8.10}$$

The rotation R (and hence the rotor R) is therefore found directly from the SVD of the function F.

9 Elasticity

The subject of the behaviour of solids under applied stress is one of the oldest in physics. Despite its great history, the subject is still rapidly evolving, driven by advances in engineering, and the advent of new materials with unusual properties. Here we review how the combination of linear algebra and geometric calculus is applied to the subject of elasticity. We show how arbitrary, nonlinear strains can be handled, before reducing to the simpler linearised theory. We also look at the behaviour of an elastic filament, which is the simplest system to extend to the nonlinear regime.

9.1 The Displacement Field

The central idea in treating elastic deformations is essentially the same as that used in rigid body dynamics. We imagine an undeformed, reference configuration and denote a position in this with the vector $\boldsymbol{x}$. Each point in the reference configuration maps to a point $\boldsymbol{y}$ in the physical configuration. The map between these is a function of position and time, which we write as

$$\boldsymbol{y} = f(\boldsymbol{x}, t). \tag{9.1}$$

(See Figure 7.) Now consider two points $\boldsymbol{x}$ and $\boldsymbol{x} + \epsilon\boldsymbol{a}$, close together in the reference configuration. The distance between these is

$$|\boldsymbol{x} + \epsilon\boldsymbol{a} - \boldsymbol{x}| = \epsilon|\boldsymbol{a}|. \tag{9.2}$$

The images of these two points in space are, dropping the time dependence, $f(\boldsymbol{x})$ and $f(\boldsymbol{x} + \epsilon\boldsymbol{a})$. The vector between these is, to first order in ϵ,

$$f(\boldsymbol{x} + \epsilon\boldsymbol{a}) - f(\boldsymbol{x}) = \epsilon\boldsymbol{a}\cdot\boldsymbol{\nabla} f(\boldsymbol{x}). \tag{9.3}$$

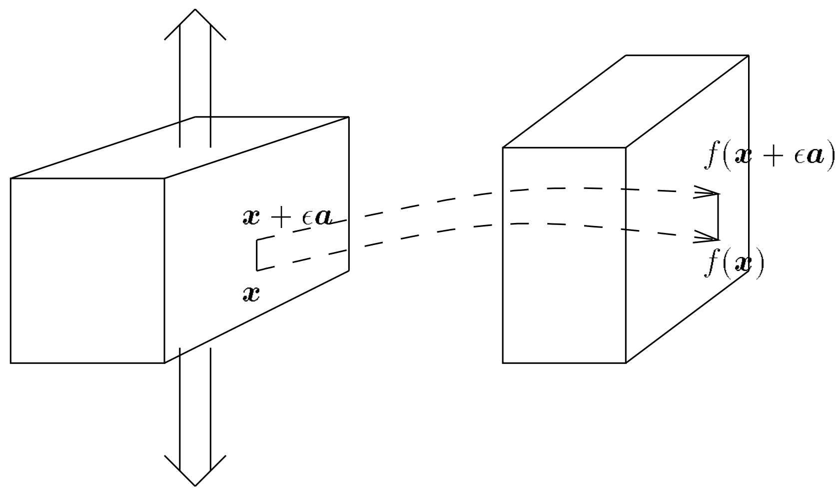


Figure 7: *An Elastic Deformation.* The nonlinear function $f(\boldsymbol{x})$ maps a point in the reference configuration to a point in space. The directional derivatives of $f(\boldsymbol{x})$ tell us about the strains in the material.

The directional derivatives of $f(\boldsymbol{x},t)$ therefore contain information about the local distortion of the material. This information is summarised in the linear function

$$\mathsf{f}(\boldsymbol{a}) = \mathsf{f}(\boldsymbol{a};\boldsymbol{x},t) = \boldsymbol{a}\cdot\boldsymbol{\nabla} f(\boldsymbol{x},t). \tag{9.4}$$

This is a time-dependent linear function of $\boldsymbol{a}$, defined for each point $\boldsymbol{x}$ in the reference configuration. One way to think of the function $\mathsf{f}(\boldsymbol{a})$ is as follows: suppose that the material is filled with a series of curves (these could be realised physically using dyes in the formation process, like in a multicoloured eraser). If the tangent vector to one of these curves in the undistorted medium is given by the vector $\boldsymbol{a}$ then, after the distortion, this vector transforms to $\mathsf{f}(\boldsymbol{a})$.

The distance between the images of the points $\boldsymbol{x}$ and $\boldsymbol{x}+\epsilon\boldsymbol{a}$ is now

$$\epsilon|\mathsf{f}(\boldsymbol{a})| = \epsilon\surd(\mathsf{f}(\boldsymbol{a})^2). \tag{9.5}$$

The $\mathsf{f}(\boldsymbol{a})^2$ term can be written as

$$\mathsf{f}(\boldsymbol{a})^2 = \langle \mathsf{f}(\boldsymbol{a})\mathsf{f}(\boldsymbol{a})\rangle = \langle \boldsymbol{a}\bar{\mathsf{f}}\mathsf{f}(\boldsymbol{a})\rangle = \boldsymbol{a}\cdot\bar{\mathsf{f}}\mathsf{f}(\boldsymbol{a}), \tag{9.6}$$

where $\bar{\mathsf{f}}$ is the adjoint (transpose) of the linear function f. The function $\mathsf{g} = \bar{\mathsf{f}}\mathsf{f}$ is therefore responsible for the change in distance between points in the undistorted and distorted medium. This must therefore be directly related to the *strain tensor* for the solid. The strain in the undistorted medium should be zero, which corresponds to f being the identity. A suitable definition for the strain tensor $\mathcal{E}(\boldsymbol{a})$ is therefore

$$\mathcal{E}(\boldsymbol{a}) = \tfrac{1}{2}(\mathsf{g}(\boldsymbol{a}) - \boldsymbol{a}). \tag{9.7}$$

The factor of 1/2 is included so that the strain tensor has the correct linearisation properties.

An alternative definition of the strain tensor, which has a number of features to recommend it, is

$$\mathcal{E}(\boldsymbol{a}) = \tfrac{1}{2}\ln \mathsf{g}(\boldsymbol{a}). \tag{9.8}$$

To date, this alternative definition has not been seriously considered. One advantage of this choice is that

$$\mathrm{Tr}(\mathcal{E}) = \ln(\det \mathsf{f}) \tag{9.9}$$

so the trace of the strain tensor is directly related to the volume scale factor. Which of the possible definitions of the strain tensor should be used can ultimately only be settled by the accuracy of the predictions of models based on the different choices.

The strain tensor is symmetric and tells us about the strains in the distorted body. The function $\mathcal{E}(\boldsymbol{a})$ takes as its argument a vector in the fixed, undistorted medium, and it returns a vector in the same medium. As with rigid body dynamics, this turns out to be the easiest way to work. The function $\mathsf{g}(\boldsymbol{a})$ is symmetric, which ensures that any overall rotational component of the strain has been factored out. $\mathsf{g}(\boldsymbol{a})$ is also positive definite, so its properties are most naturally discussed in terms of its eigenvectors. These are the directions in the reference body which are stretched, but not rotated, for a given strain.

9.2 Stress and the Balance Equations

The contact force between two surfaces in the medium is a function of the normal to the surface (and position and time). Cauchy showed that, given suitable continuity conditions, the force must be a linear function of the normal. We write this as $T(\boldsymbol{n})$. Since $T(-\boldsymbol{n}) = -T(\boldsymbol{n})$ it follows that Newton's third law is immediately satisfied. The function $T(\boldsymbol{n})$ takes as its argument a vector in the reference configuration, and returns a vector in the material body (see Fig. 8). We need a means to pull this back to the reference configuration, so that the stress can be related to the strain there. To see how to do this we need to consider the balance equations (*i.e.* force laws) in the material body.

The total force on an element of volume V is found by integrating $T(\boldsymbol{n})$ over the surface of the element, so we have

$$\int \rho \frac{\partial^2 \boldsymbol{y}}{\partial t^2}\, dV = \oint T(d\boldsymbol{s}), \tag{9.10}$$

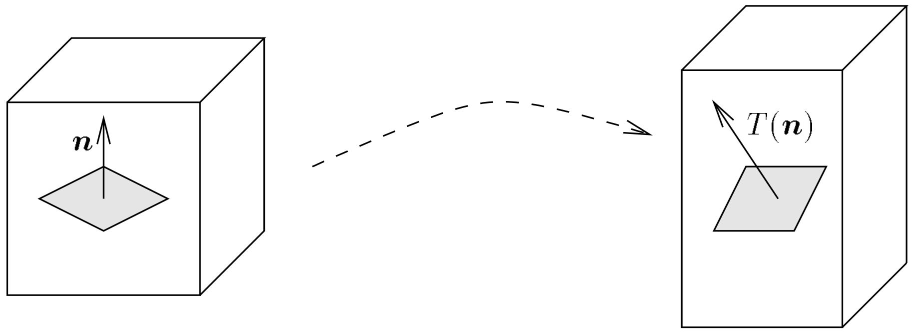


Figure 8: *The Stress.* The stress tensor $T(\boldsymbol{n})$ returns the force in the material body over the plane with normal $\boldsymbol{n}$, in the reference body.

where $\rho = \rho(\boldsymbol{x})$ is the density in the undistorted medium. A simple application of the divergence theorem converts the surface integral to a volume integral,

$$\oint T(d\boldsymbol{s}) = \int T(\overleftarrow{\boldsymbol{\nabla}})\, dV \tag{9.11}$$

from which we can read off the dynamical equation

$$\rho \frac{\partial^2 \boldsymbol{y}}{\partial t^2} = T(\overleftarrow{\boldsymbol{\nabla}}). \tag{9.12}$$

The second balance equation is found by considering the total couple on a volume element, and relating this to the change in angular momentum. The total couple about the point $\boldsymbol{y}_0$ is

$$M = \oint (\boldsymbol{y} - \boldsymbol{y}_0) \wedge T(d\boldsymbol{s}). \tag{9.13}$$

This must be equated with the change in angular momentum,

$$\frac{\partial}{\partial t} \int \rho(\boldsymbol{y} - \boldsymbol{y}_0) \wedge \dot{\boldsymbol{y}}\, dV = \int (\boldsymbol{y} - \boldsymbol{y}_0) \wedge T(\overleftarrow{\boldsymbol{\nabla}})\, dV. \tag{9.14}$$

Applying the divergence theorem again, we find that angular momentum balance is satisfied provided

$$\partial_i \boldsymbol{y} \wedge T(\boldsymbol{e}_i) = \mathsf{f}(\boldsymbol{e}_i) \wedge T(\boldsymbol{e}_i) = 0. \tag{9.15}$$

It follows that the tensor $\mathcal{T}(\boldsymbol{n})$ is symmetric, where

$$\mathcal{T}(\boldsymbol{n}) = \mathsf{f}^{-1} T(\boldsymbol{n}). \tag{9.16}$$

This is the (first) Piola-Kirchoff stress tensor. $\mathcal{T}(\boldsymbol{n})$ is a symmetric tensor defined entirely in the reference configuration, since the f^{-1} term maps the vector $T(\boldsymbol{n})$ back to the reference copy. The Piola-Kirchoff tensor is the one that we must relate to the strain tensor, via the constitutive relations of the material.

9.3 Constitutive Equations

The strain tensor can, in principle, be a quite general function of the applied stresses. Complications can include a lack of homogeneity and isotropy, viscosity, thermal and chemical properties, and a dependence on the history of the body. For a wide range of applications, however, we can restrict to the simplest case of a *linear, isotropic, homogeneous* (LIH) body. In these the stresses and strains are related linearly by just two parameters, the bulk modulus B and the shear modulus G. For LIH media the relation between the applied stress $\mathcal{T}(\boldsymbol{a})$ and the strain $\mathcal{E}(\boldsymbol{a})$ is

$$\mathcal{T}(\boldsymbol{a}) = 2G\mathcal{E}(\boldsymbol{a}) + (B - \tfrac{2}{3}G)\mathrm{Tr}(\mathcal{E})\boldsymbol{a}. \tag{9.17}$$

The bulk modulus B describes how the body responds to isotropic pressure, as is the case when the body is immersed in a liquid. The applied stress is then a uniform pressure P in all directions, so we have

$$\mathcal{T}(\boldsymbol{a}) = -P\boldsymbol{a}. \tag{9.18}$$

The sign is negative, because the force is a compression, rather than a stretch. Taking the trace of both sides of equation (9.17) gives

$$-3P = 3B\mathrm{Tr}(\mathcal{E}). \tag{9.19}$$

The distortion in the medium will be given by

$$f(\boldsymbol{x}) = \lambda\boldsymbol{x} + \boldsymbol{x}_0, \tag{9.20}$$

where $\boldsymbol{x}_0$ is the vector from the origin to its image in the physical configuration, and λ is the scale factor. It follows that

$$\mathsf{f}(\boldsymbol{a}) = \lambda\boldsymbol{a}, \tag{9.21}$$

and hence $\mathrm{Tr}(\mathcal{E}) = 3(\lambda^2 - 1)/2$. The bulk modulus is then given by

$$B = -\frac{2P}{3(\lambda^2 - 1)}. \tag{9.22}$$

If we now linearise by setting $\lambda = 1 + \epsilon$, we recover the familiar result that

$$B = -\frac{P}{3\epsilon} = -P\frac{V}{\Delta V} \tag{9.23}$$

where V is the volume. Since the force is a compression, the change in volume ΔV will be negative.

When a stress is applied along a single direction, the body will respond by stretching along the direction of the applied force, and contracting in the other two directions. The relative sizes of these effects is controlled by the shear modulus G — the second of the two main elastic parameters. Given a set of constitutive relations and the balance equations, one has enough information to compute the evolution of the system. The resulting equations are, in general, highly complicated and nonlinear, even if the material itself is linear. For this reason it is usual to work in the linear regime of small deformations whenever possible.

9.4 Linearised Elasticity

Suppose that the elastic deformation can be written as

$$\boldsymbol{x}' = f(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{x}_0 + \boldsymbol{\epsilon} \tag{9.24}$$

where $\boldsymbol{\epsilon}$ is a vector field. The directional derivatives of this are denoted with an underbar, so

$$\underline{\epsilon}(\boldsymbol{a}) = \boldsymbol{a}\cdot\boldsymbol{\nabla}\boldsymbol{\epsilon}. \tag{9.25}$$

Working to first order, the strain tensor becomes

$$\mathcal{E}(\boldsymbol{a}) = \tfrac{1}{2}\big(\underline{\epsilon}(\boldsymbol{a}) + \overline{\epsilon}(\boldsymbol{a})\big). \tag{9.26}$$

The stress tensor $\mathcal{T}(\boldsymbol{a})$ gives the applied force over the surface perpendicular to $\boldsymbol{a}$ in the reference configuration. In the linearised theory, this is the same as the force in the material body (to first order). Assuming an LIH material, we then recover the dynamical equations

$$G\boldsymbol{\nabla}^2\boldsymbol{\epsilon} + (B + \tfrac{1}{3}G)\boldsymbol{\nabla}\,\boldsymbol{\nabla}\cdot\boldsymbol{\epsilon} = \rho\frac{\partial^2\boldsymbol{\epsilon}}{\partial t^2}. \tag{9.27}$$

For many applications we assume a harmonic time variation $\cos(\omega t)$, for which we recover the *vector Helmholtz equation*,

$$v_l^2\boldsymbol{\nabla}\,\boldsymbol{\nabla}\cdot\boldsymbol{\epsilon} + v_t^2\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\wedge\boldsymbol{\epsilon}) + \omega^2\boldsymbol{\epsilon} = 0. \tag{9.28}$$

Here the equation has been expressed in terms of longitudinal and transverse sound speeds v_l and v_t, given by

$$v_l^2 = \frac{B + \frac{4}{3}G}{\rho}, \qquad v_t^2 = \frac{G}{\rho}. \tag{9.29}$$

The vector Helmholtz equation is used to study many phenomena, ranging from oscillations of an elastic sphere to the propagation of waves created by an earthquake.

9.5 The Elastic Filament

We now treat the bending and twisting of an elastic filament under static loads. Suppose that the filament is described by the curve $\boldsymbol{x}(\lambda)$. We will choose λ to be the affine parameter along the curve, so that $\boldsymbol{x}' = \partial_\lambda\boldsymbol{x}$ is a unit vector. This vector can be identified with the third vector of an orthonormal frame,

$$\boldsymbol{x}' = \boldsymbol{f}_3 = R(\lambda)\boldsymbol{e}_3R^\dagger(\lambda). \tag{9.30}$$

The remaining two vectors then determine two directions perpendicular to the filament, and can be used to describe any internal twisting in the filament. With this approach, both the bending and twisting of the filament are described in the single equation for the rotor R.

A thin beam or filament has stiffness to bending. When it is bent, a bending moment (couple) is set up which is linearly related to the curvature. In terms of the two principal directions in the filament, the appropriate formula for the bending moment is

$$M = \frac{YI}{R}, \tag{9.31}$$

where Y is Young's modulus, I is the relevant principal moment of area, and R is the radius of curvature in the plane of the bending. The radius of curvature is determined by the magnitude of the projection of the vector $\boldsymbol{f}_3'$ into the relevant plane. So the radius of curvature in the $\boldsymbol{f}_1\boldsymbol{f}_3$ plane, for example, is given by

$$\frac{1}{R_1} = |\boldsymbol{f}_3' \cdot (\boldsymbol{f}_1\boldsymbol{f}_3)\boldsymbol{f}_3\boldsymbol{f}_1| = |\boldsymbol{f}_1 \cdot \boldsymbol{f}_3'|. \tag{9.32}$$

To compute $\boldsymbol{f}_3'$ we first need to establish an important result for the derivative of a rotor. Rotors are normalised to unity, so $RR^\dagger = 1$. Differentiating this, we obtain

$$R'R^\dagger + RR^{\dagger\prime} = 0. \tag{9.33}$$

It follows that

$$R'R^\dagger = -RR^{\dagger\prime} = -(R'R^\dagger)^\dagger. \tag{9.34}$$

The quantity $R'R^\dagger$ is therefore equal to minus its reverse (and has even grade) so must be a pure bivector. We set this equal to $-\Omega/2$, so that

$$R' = -\tfrac{1}{2}\Omega R = -\tfrac{1}{2}R\Omega_B, \tag{9.35}$$

where $\Omega_B = R^\dagger \Omega R$. It follows that

$$\boldsymbol{f}_3' = R'\boldsymbol{e}_3R^\dagger + R\boldsymbol{e}_3R^{\dagger\prime} = \tfrac{1}{2}(-\Omega\boldsymbol{f}_3 + \boldsymbol{f}_3\Omega) = \boldsymbol{f}_3 \cdot \Omega, \tag{9.36}$$

so the radius of curvature just picks out one coefficient of Ω.

Equation (9.31) can correspondingly be used to find the curvature induced by an applied couple C. With C and Ω given in terms of components by

$$C = \sum_k c_k I\boldsymbol{f}_k, \qquad \Omega = \sum_k \omega_k I\boldsymbol{f}_k, \tag{9.37}$$

we find that the curvature and the couple are related by

$$c_1 = Y i_1 \omega_1, \qquad c_2 = Y i_2 \omega_2, \tag{9.38}$$

where i_1 is the moment of area measured perpendicular to the $\boldsymbol{f}_1$ direction.

In addition to its stiffness to bending, the filament has a stiffness to torsion. For the case of elastic behaviour, the twist in the $\boldsymbol{f}_1 \boldsymbol{f}_2$ plane is proportional to the applied torque, and we have

$$c_3 = G i_3 \boldsymbol{f}_1' \cdot \boldsymbol{f}_2 = G i_3 \omega_3. \tag{9.39}$$

The applied couple C and the 'curvature bivectors' Ω and Ω_B are therefore related by

$$C = Y(i_1\omega_1 \, I\boldsymbol{f}_1 + i_2\omega_2 \, I\boldsymbol{f}_2) + G i_3 \omega_3 \, I\boldsymbol{f}_3 = R\mathcal{I}(\Omega_B)R^\dagger, \tag{9.40}$$

which defines the linear function $\mathcal{I}$ (which maps bivectors to bivectors). We can invert this relation to give

$$\Omega_B = \mathcal{I}^{-1}(R^\dagger C R), \tag{9.41}$$

which expresses the curvature bivector Ω_B in terms of the applied couple C and the elastic constants. The full set of equations are now (9.30) and (9.41), together with the rotor equation

$$\frac{dR}{d\lambda} = -\tfrac{1}{2} R(\Omega_B + \Omega_0), \tag{9.42}$$

where the bivector Ω_0 expresses the natural shape of the filament.

An advantage of this set of equations is that locally small distortions of the filament can be allowed to build up into large, global deviations. An interesting simple case is that of a *wrench*, where

$$C(\boldsymbol{x}) = C_0 + \boldsymbol{f} \wedge \boldsymbol{x}, \tag{9.43}$$

where C_0 and $\boldsymbol{f}$ are respectively the couple and force applied at the ends. A wrench such as this describes the general case of a light filament loaded at its ends. Figure 9 shows the type of distortion that can result.

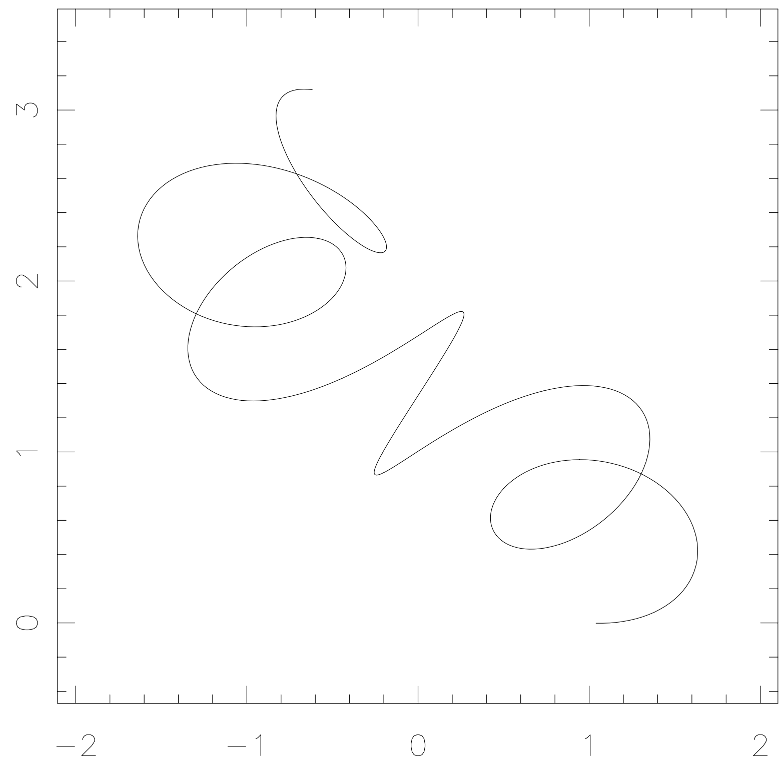


Figure 9: *A filament loaded at its two ends.* Two directions are shown, though there is also considerable structure in the third. The material has $i_1 = i_2$ and a zero Poisson ratio.

10 Relevant Papers

The following list of papers, courses, and notes discuss in detail some of the applications outlined in the talks.

C.J.L. Doran and A.N. Lasenby, Lecture Notes to accompany 4th year undergraduate course on *Physical Applications of Geometric Algebra*, 2000.
Available at `http://www.mrao.cam.ac.uk/~clifford/ptIIIcourse/`.

David Hestenes, *New Foundations for Classical Mechanics* (Second Edition).
Published by Kluwer Academic.

J. Lasenby and A. Stevenson. Using geometric algebra in optical motion capture.
In E. Bayro and G. Sobczyk, editors, *Geometric algebra: A geometric approach to computer vision, neural and quantum computing, robotics and engineering.*

L. Dorst, S. Mann and T. Bouma. *GABLE: A Matlab Tutorial for Geometric Algebra.*
Available at `www.carol.wins.uva.nl/~leo/clifford/gablebeta.html`

C.J.L. Doran. Bayesian inference and geometric algebra: an application to camera localization.
In E. Bayro and G. Sobczyk, editors, *Geometric algebra: A geometric approach to computer vision, neural and quantum computing, robotics and engineering.*

J. Lasenby, W.J. Fitzgerald, A.N. Lasenby and C.J.L. Doran. New geometric methods for computer vision – an application to structure and motion estimation.
International Journal of Computer Vision, 26(3), 191-213. 1998.

J. Clements. 1999 *Beam buckling using geometric algebra.*
M.Eng. project report, Cambridge University Engineering Department.

L. Dorst. Honing geometric algebra for its use in the computer sciences.
In G. Sommer, editor, *Geometric Computing with Clifford Algebras.* Springer.

M. Ringer and J. Lasenby, 2000 *Modelling and tracking articulated motion from multiple camera views.*
Cambridge University Engineering Department Report CUED/F-INFENG/-TR.378.

Geo-Metric-Affine-Projective Computing

Speaker: Ambjörn Naeve, Ph.D., Senior researcher

Affiliation: Centre for user-oriented IT-Design (CID)
Dept. of Numerical Analysis and Computing Science
Royal Institute of Technology (KTH)
100 44 Stockholm, Sweden

email-address: amb@nada.kth.se

web-site: cid.nada.kth.se/il

SIGGRAPH 2001 CREATE INTERACTION

Projective Drawing Board (PDB)

PDB is an interactive program for doing plane projective geometry that will be used to illustrate this lecture.

PDB has been developed by
Harald Winroth and ***Ambjörn Naeve***
as a part of Harald's doctoral thesis work at
the Computational Vision and Active Perception
(CVAP) laboratory at KTH .

PDB is avaliable as freeware under Linux.

www.nada.kth.se/~amb/pdb-dist/linux/pdb2.5.tar.gz

H. Grassmann

Yours most truly

W. K. Clifford

Geometric algebra in *n*-dim Euclidean space

Underlying vector space $\mathbf{V}^n$ with ON-basis $e_1,\ldots,e_n$.

Geometric algebra: $\mathbf{G} = \mathbf{G}_n \equiv \mathbf{G}(\mathbf{V}^n)$ has 2^n dimensions.

A *multivector* is a sum of *k-vectors*: $M = \sum_{k=0}^{n} \langle M \rangle_k$

A *k-vector* is a sum of *k-blades*: $\langle M \rangle_k = A_k + B_k + \ldots$

A *k-blade* = blade of *grade k* : $B_k = b_1 \wedge b_2 \wedge \ldots \wedge b_k$

Note: $B_k \neq 0 \Leftrightarrow b_1,\ldots,b_k$ are linearly independent.

Hence: the grade of a blade is the dimension of the subspace that it spans.

Blades correspond to geometric objects

blade of grade	equivalence class of directed	equal orientation and
1	line segments	length
2	surface regions	area
3	3-dim regions	volume
$\vdots$	$\vdots$	$\vdots$
k	k-dim regions	k-volume

Pseudoscalars and duality

Def: A n-blade in $\mathbf{G}_n$ is called a ***pseudoscalar***.

A pseudoscalar: $P = p_1 \wedge p_2 \wedge \ldots \wedge p_n$

A *unit* pseudoscalar: $I = e_1 \wedge e_2 \wedge \ldots \wedge e_n$

The *bracket* of P: $[P] = PI^{-1}$

The *dual* of a multivector x: $\mathrm{Dual}(x) = xI^{-1}$

Notation: $\mathrm{Dual}(x) = x^*$

Note: If A is a k-blade, then A^* is a (n-k)-blade.

SIGGRAPH 2001 CREATE INTERACTION

The subspace of a blade

Fact: To every non-zero *m*-blade $B = b_1 \wedge \ldots \wedge b_m$

there corresponds a *m*-dim subspace $\overline{B} \subset \mathbf{V}^n$

with $\overline{B} = \text{Linspan}\{b_1, \ldots, b_m\}$

$$= \text{Linspan}\{b \in \mathbf{V}^n : b \wedge B = 0\} .$$

Fact: If $e_1, e_2, \ldots, e_m$ is an ON-basis for $\overline{B}$

and if $b_i = \sum_{k=1}^{m} b_{ik} e_k$ for $i = 1,\ldots,m$,

then $B = \left(\det b_{ik}\right) e_1 \wedge e_2 \wedge \ldots \wedge e_m$

$$= \left(\det b_{ik}\right) e_1 e_2 \ldots e_m .$$

Dual subspaces <=> orthogonal complements

Fact: If A is a non-zero m-blade $\overline{A^*} = \overline{A}^{\perp}$.

Proof: We can WLOG choose an ON-basis for $\mathbf{V}^n$ such that

$$A = \lambda e_1 e_2 \ldots e_m \quad \text{and} \quad I = e_1 e_2 \ldots e_n .$$

We then have

$$A^* = AI^{-1} = \pm \lambda e_{m+1} \ldots e_n$$

which implies that

$$\overline{A^*} = \overline{A}^{\perp} .$$

The join and the meet of two blades

Def: Given blades A and B, if there exists a blade C such that $A = BC = B \wedge C$ we say that A is a ***dividend*** of B and B is a ***divisor*** of A.

Def: The ***join*** of blades A and B is a *common dividend of lowest grade*.

Def: The ***meet*** of blades A and B is a *common divisor of greatest grade*.

The join and meet provide a representation in geometric algebra of the ***lattice algebra*** of ***subspaces*** of $\mathbf{V}^n$.

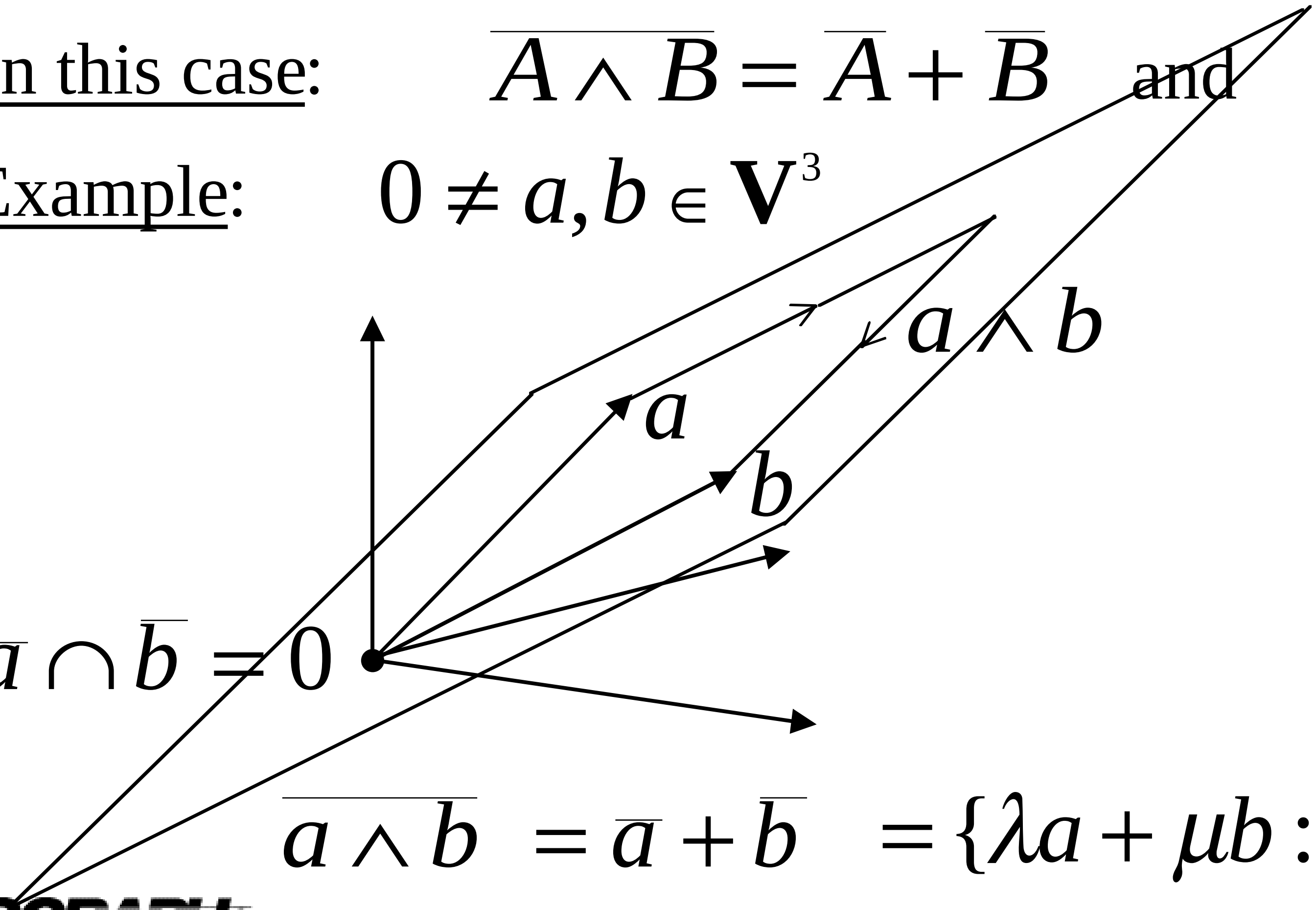
Join of two blades <=> sum of their subspaces

Def: For two blades A and B with $A \wedge B \neq 0$

we can define: $\mathrm{Join}(A, B) = A \wedge B$.

In this case: $\overline{A \wedge B} = \overline{A} + \overline{B}$ and $\overline{A} \cap \overline{B} = 0$.

Example: $0 \neq a, b \in \mathbf{V}^3$



$\overline{a} \cap \overline{b} = 0$

$\overline{a \wedge b} = \overline{a} + \overline{b} = \{\lambda a + \mu b : \lambda, \mu \in \mathrm{R}\}$

Meet of blades <=> intersection of subspaces

Def: If blades $A, B \neq 0$ and $\overline{A} + \overline{B} = \mathbf{V}^n$

then: $\mathrm{Meet}(A, B) \equiv A \vee B = (A^* \wedge B^*)I$.

In this case: $\overline{A \vee B} = \overline{A} \cap \overline{B}$.

Note: The meet product is related
to the outer product
by duality:

$$(A \vee B)I^{-1} = (A^* \wedge B^*)II^{-1} = A^* \wedge B^*$$

$$\mathrm{Dual}(A \vee B) = \mathrm{Dual}(A) \wedge \mathrm{Dual}(B)$$

Dual outer product

Dualisation:

$$\mathcal{G} \xrightarrow{*} \mathcal{G}$$
$$x \mapsto x^{*} = xI^{-1}$$

Dual outer product:

$$\begin{tikzcd}
\mathcal{G} \times \mathcal{G} \arrow[r, "\vee"] \arrow[d, "*"'] & \mathcal{G} \arrow[d, "*"] \\
\mathcal{G} \times \mathcal{G} \arrow[r, "\wedge"] & \mathcal{G}
\end{tikzcd}$$

$$x \vee y = ((xI^{-1}) \wedge (yI^{-1}))I$$

$$x^{*} \wedge y^{*} = (x \vee y)^{*}$$

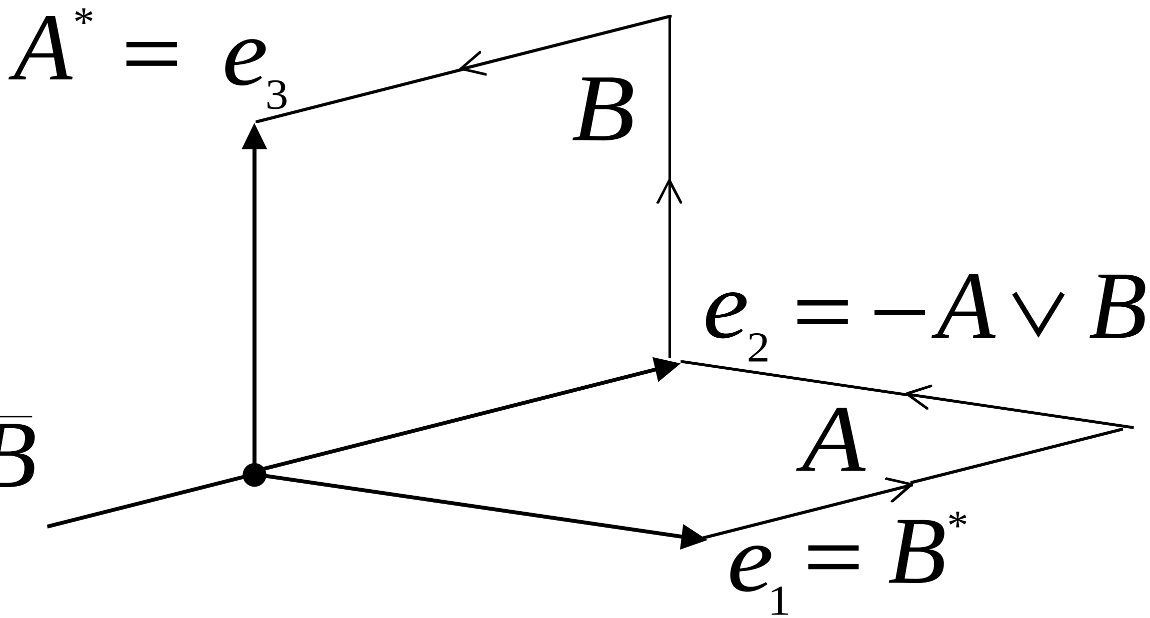
Example: $\quad \mathbf{V}^3 \quad , \quad I = e_1 \wedge e_2 \wedge e_3 = e_1 e_2 e_3$

$$A = e_1 \wedge e_2 = e_1 e_2 \quad , \quad B = e_2 \wedge e_3 = e_2 e_3$$

$$A^* = \left(e_1 \wedge e_2\right) I^{-1} = \left(e_1 e_2\right)\left(e_3 e_2 e_1\right) = (-1)^2 e_3 e_1 e_2 e_2 e_1 = e_3$$

$$B^* = \left(e_2 \wedge e_3\right) I^{-1} = \left(e_2 e_3\right)\left(e_3 e_2 e_1\right) = e_1$$

$$A \vee B = (A^* \wedge B^*) I = \left(e_3 \wedge e_1\right)\left(e_1 e_2 e_3\right) = -e_2$$

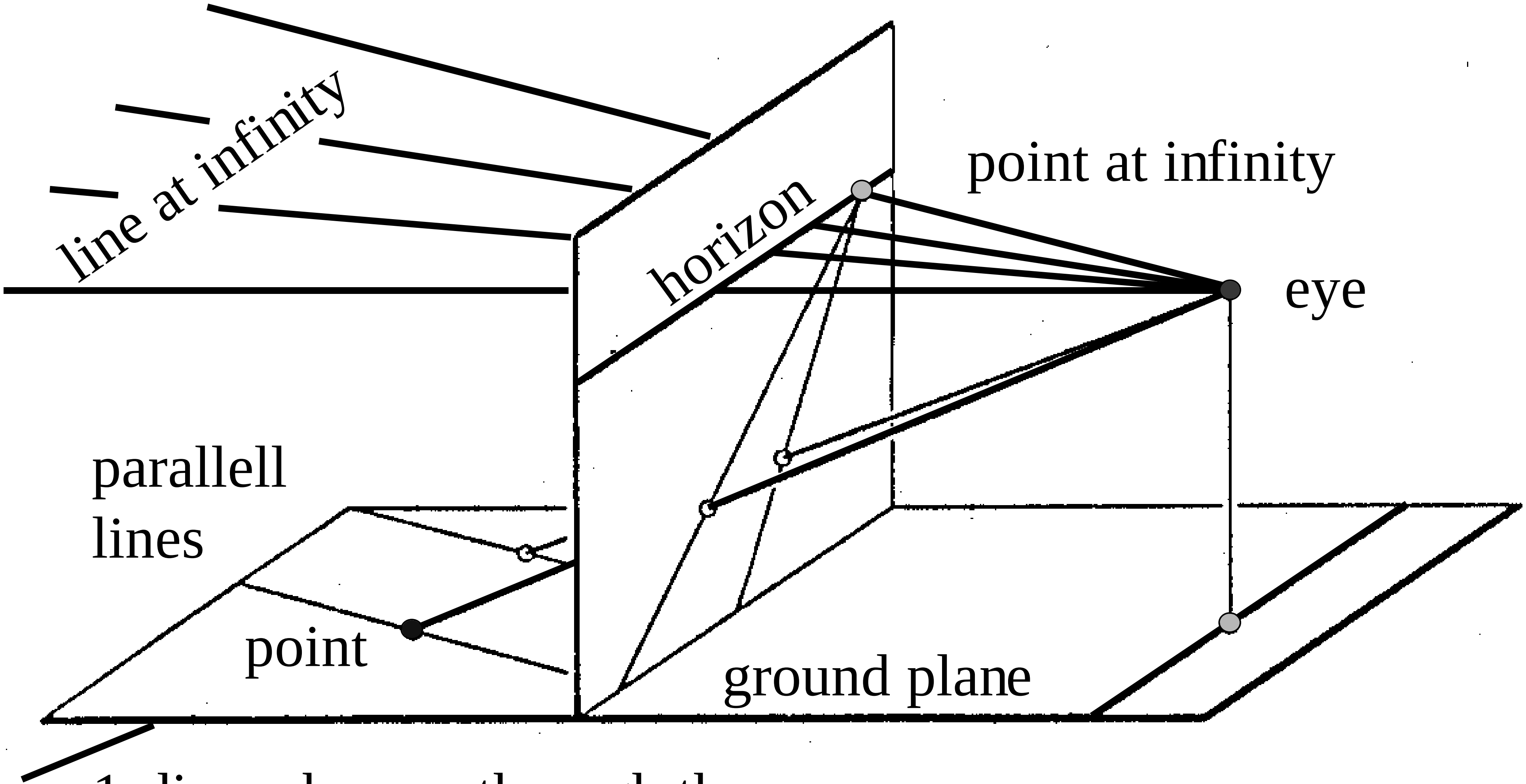


Hence:

$$\overline{A \vee B} = \overline{A} \cap \overline{B}$$

Projective geometry - historical perspective

1-d subspace parallell to the ground plane



1-dim subspace through the eye

n-dimensional projective space $\mathbf{P}^n$

$\mathbf{P}^n = \mathbf{P}(\mathbf{V}^{n+1})$ = the set of non-zero subspaces of $\mathbf{V}^{n+1}$.

A *point* p is a 1-dim subspace (spanned by a 1-blade a).

$$p = \overline{a} = \{\lambda a : \lambda \neq 0\} \doteq a \doteq \alpha a \ , \ \alpha \neq 0 .$$

A *line* l is a 2-dim subspace (spanned by a 2-blade B_2).

$$l = \overline{B_2} = \{\lambda B_2 : \lambda \neq 0\} \doteq B_2 .$$

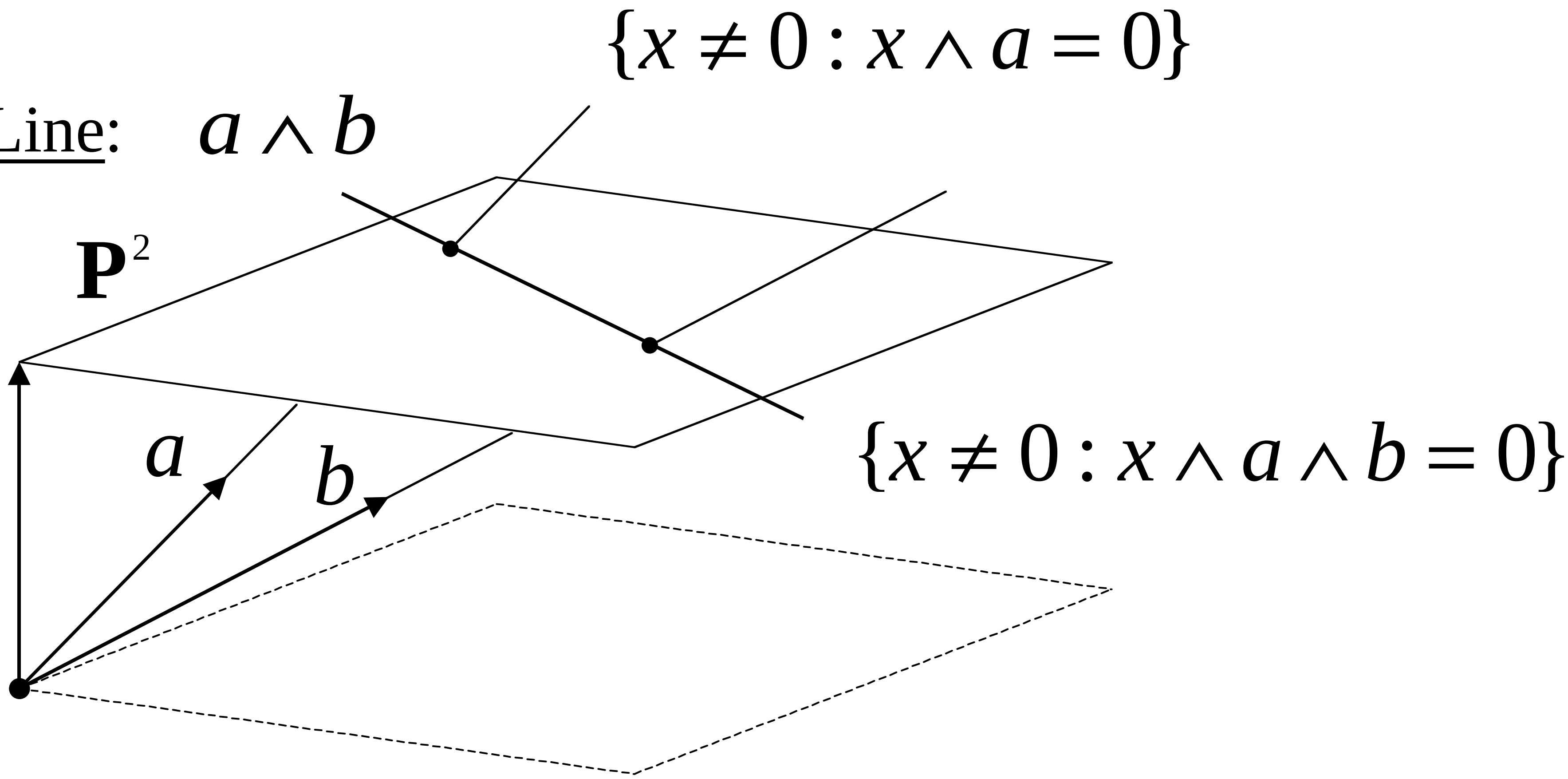
Let $\mathbf{B}$ denote the set of non-zero blades of the geometric algebra $\mathbf{G}(\mathbf{V}^{n+1})$. Hence we have the mapping

$$\mathbf{B} \ni B \longmapsto \overline{B} \in \mathbf{P}^n .$$

The projective plane $\mathbf{P}^2$

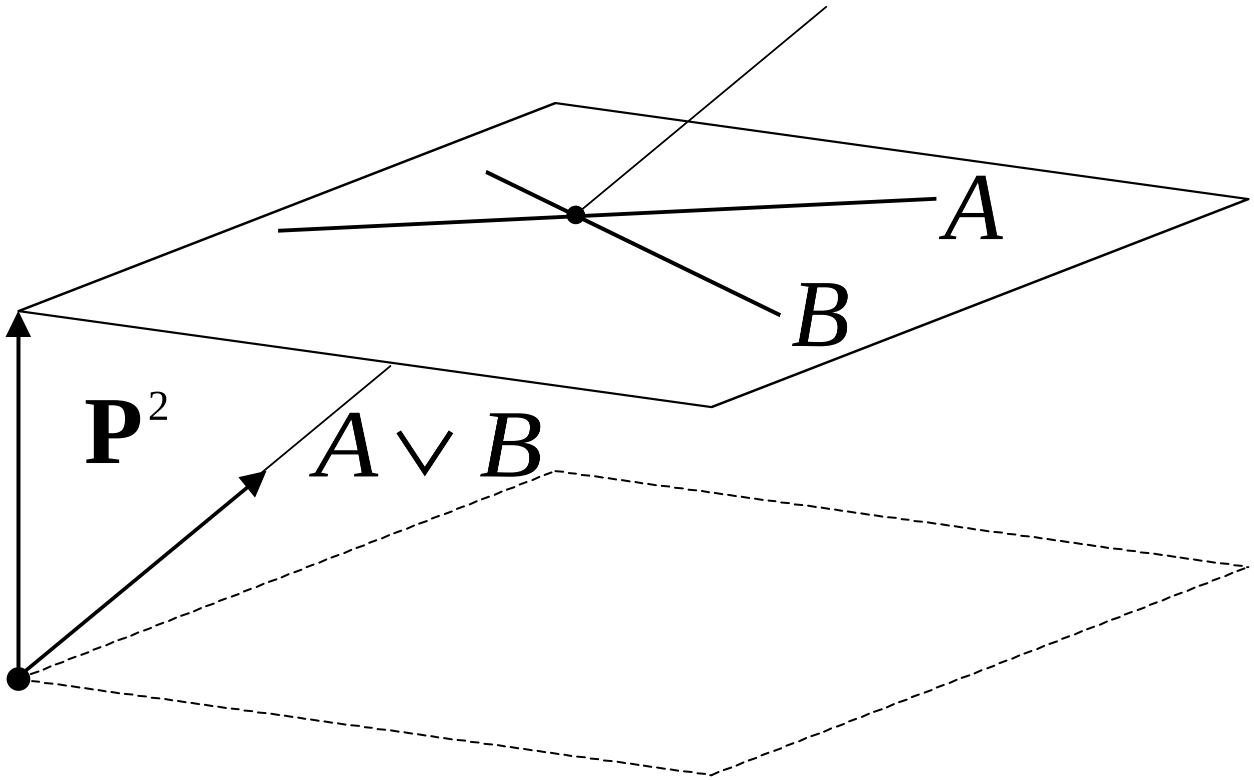
$$0 \neq a, b, x \in \mathbf{V}^3$$

Line: $a \wedge b$

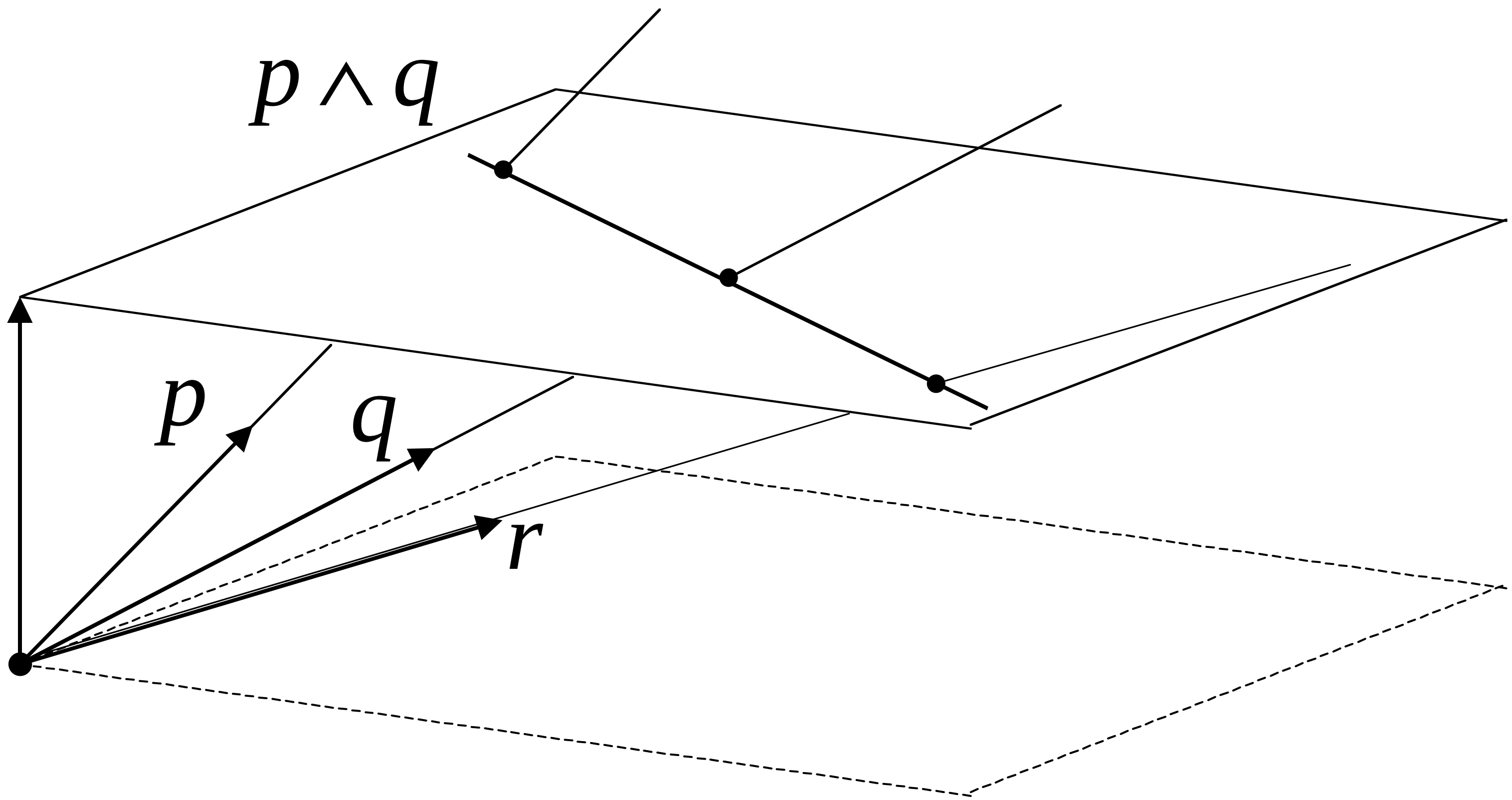


The intersection of two lines in $\mathbf{P}^2$

$$A, B \in \{\text{non-zero 2-blades in } \mathbf{G}_3\} \quad .$$



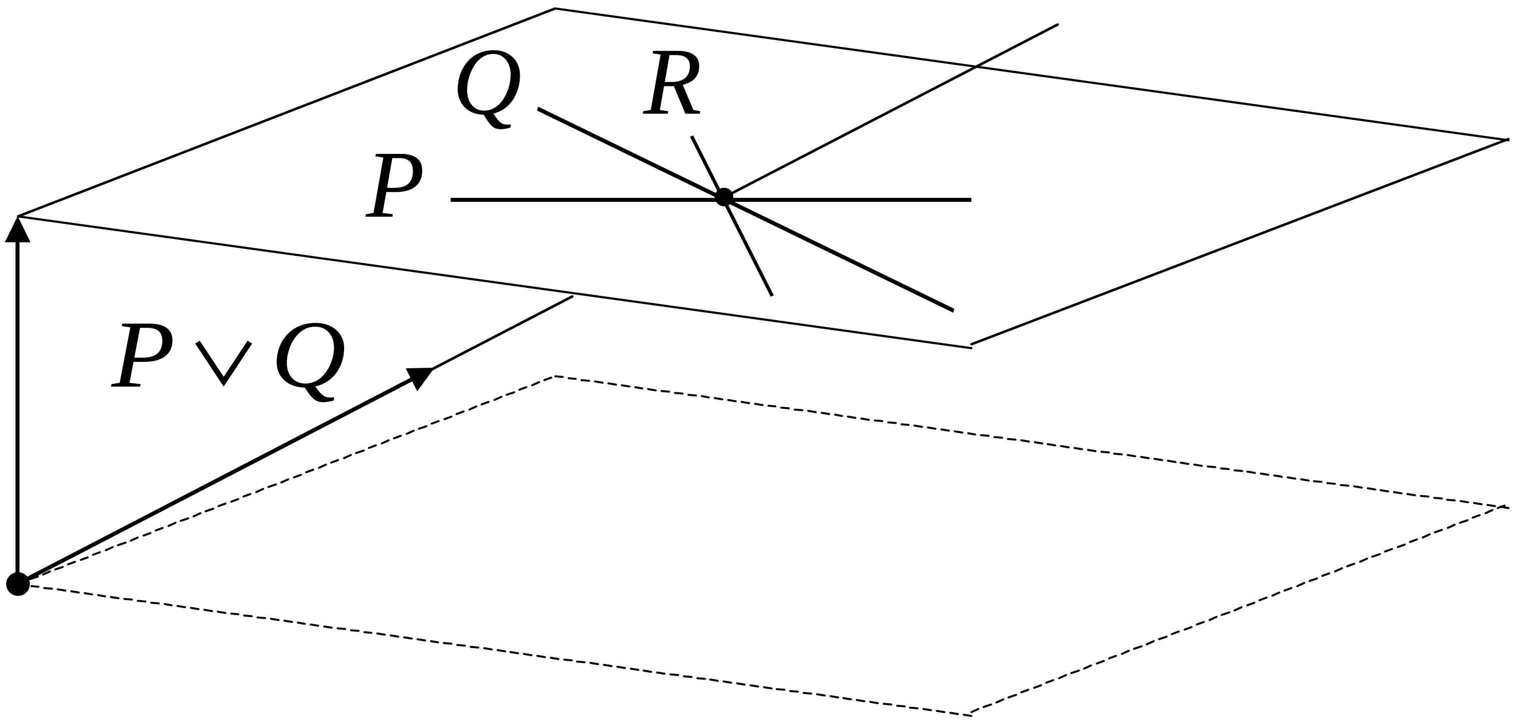
Collinear points



The points p, q, r are *collinear*

if and only if $\quad p \wedge q \wedge r = 0.$

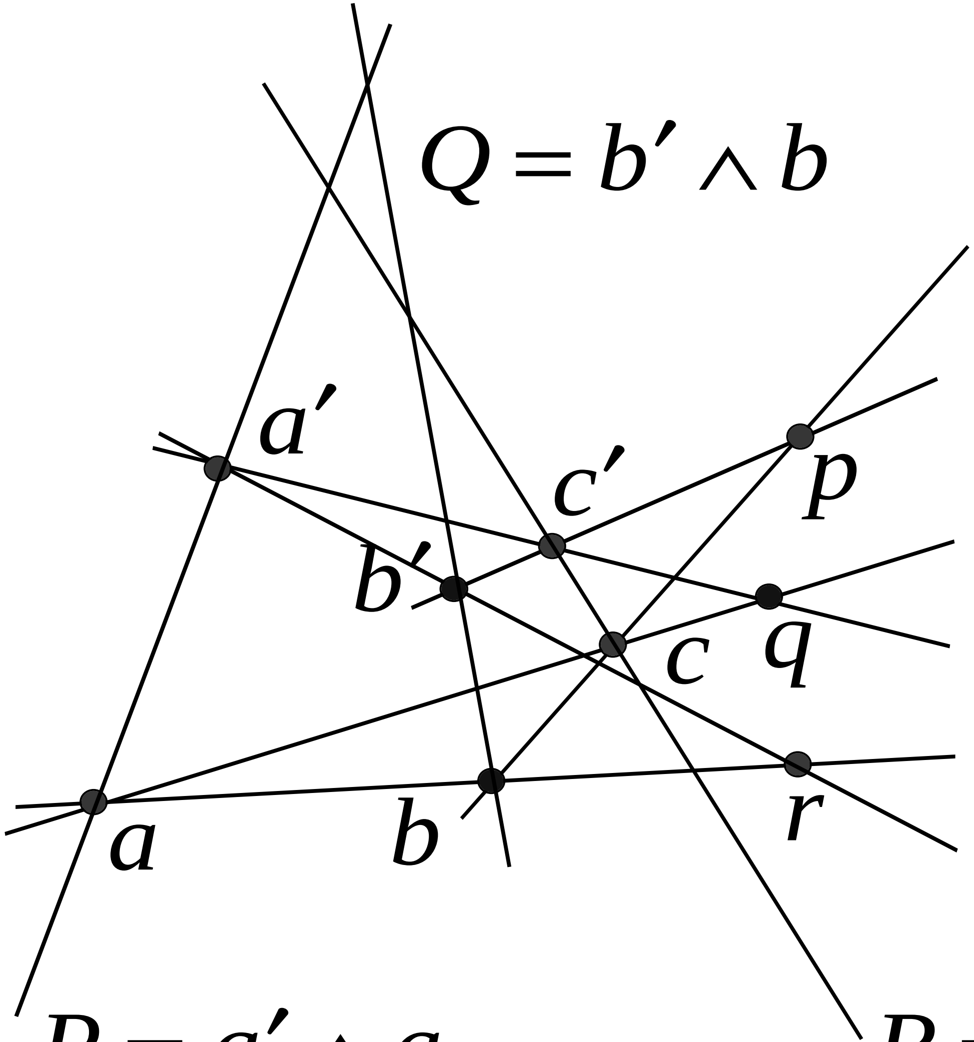
Concurrent lines



The lines P, Q, R are *concurrent*

if and only if $(P \vee Q) \wedge R = 0$.

Desargues' configuration



$$Q = b' \wedge b$$

$$A = b \wedge c \qquad p = A \vee A'$$

$$A' = b' \wedge c'$$

$$B = c \wedge a \qquad q = B \vee B'$$

$$B' = c' \wedge a'$$

$$C = a \wedge b \qquad r = C \vee C'$$

$$C' = a' \wedge b'$$

$$P = a' \wedge a \qquad R = c' \wedge c$$

Desargues' configuration (cont.)

$$P = a' \wedge a \qquad p = A \vee A' = (b \wedge c) \vee (b' \wedge c')$$

$$Q = b' \wedge b \qquad q = B \vee B' = (c \wedge a) \vee (c' \wedge a')$$

$$R = c' \wedge c \qquad r = C \vee C' = (a \wedge b) \vee (a' \wedge b')$$

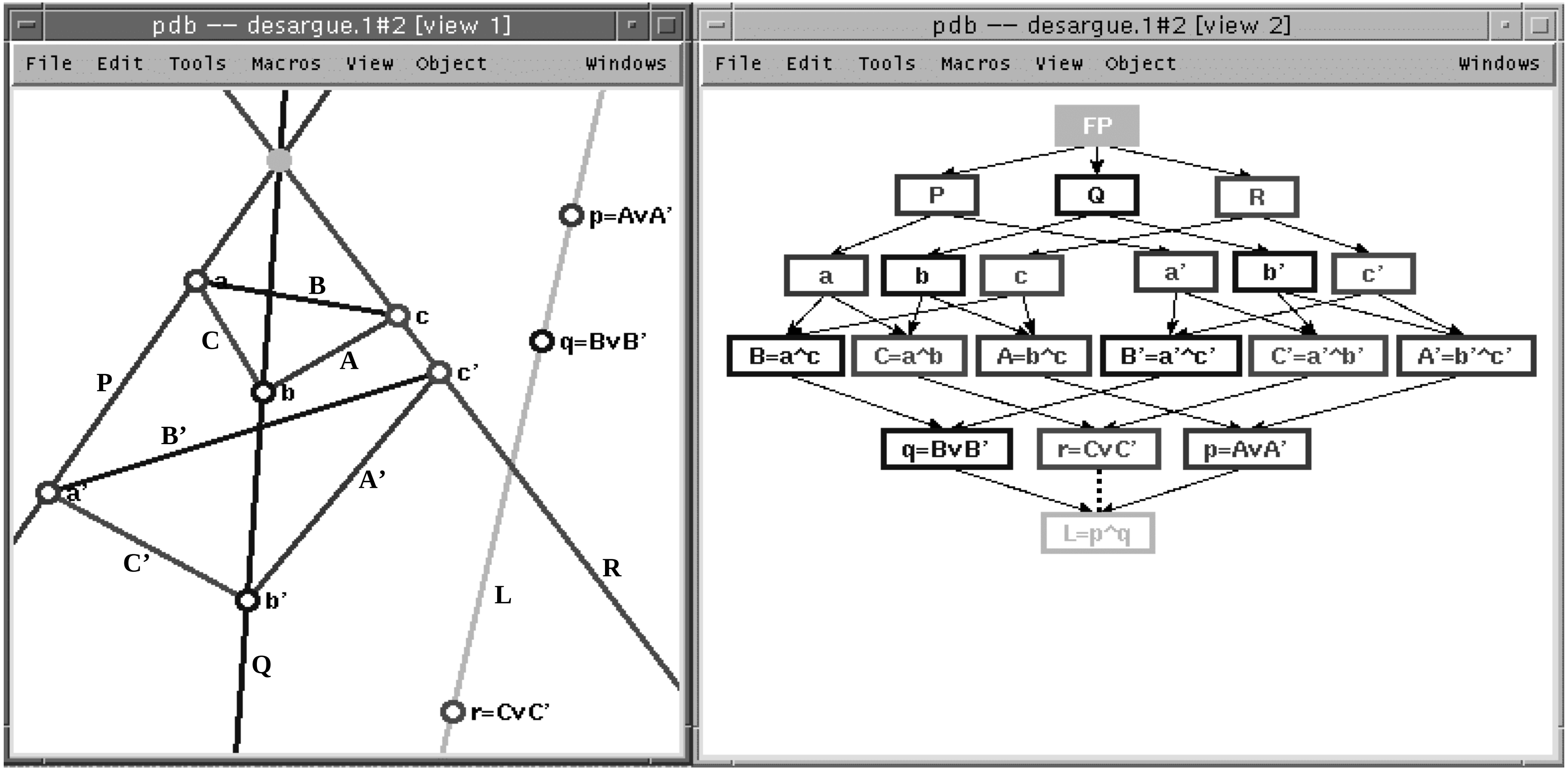
$$J = a \wedge b \wedge c = [abc]I$$

$$J' = a' \wedge b' \wedge c' = [a'b'c']I$$

Leads to:

$$\underbrace{p \wedge q \wedge r}_{\text{= 0 if and only if } p,q,r \text{ are collinear}} \equiv JJ' \underbrace{(P \vee Q) \wedge R}_{\text{= 0 if and only if } P,Q,R \text{ are concurrent}}$$

Desargues' theorem



p, q, r are collinear iff P, Q, R are concurrent.

Pascal's theorem

Let a, b, c, a', b' be five given points in $\mathbf{P}^2$.

Consider the second degree polynomial given by

$$
\begin{aligned}
p(x) = {} & ((a \wedge b') \vee (a' \wedge b)) \wedge \\
& ((b \wedge x) \vee (b' \wedge c)) \wedge \\
& ((c \wedge a') \vee (x \wedge a)) \quad .
\end{aligned}
$$

It is obvious that $p(a) = p(b) = 0$

and easy to verify that $p(c) = p(a') = p(b') = 0$.

Hence: $p(x) = 0$ must be the equation of the conic on the 5 given points.

Verifying that $p(a') = 0$

$$p(a') =$$

same line

$$= \underbrace{((a \wedge b') \vee \overbrace{(a' \wedge b)})) \wedge ((\overbrace{b \wedge a'}) \vee (b' \wedge c)) \wedge}_{\doteq a' \wedge b}$$

same point

$$\underbrace{((c \wedge a') \vee (a' \wedge a))}_{\doteq a'} \quad \doteq (a' \wedge b) \wedge a' \quad = 0$$

<u>Pascal's theorem (cont.)</u>

Hence, a sixth point c' lies on this conic if and only if

$$\begin{aligned} p(c') = ((a \wedge b') \vee (a' \wedge b)) \wedge \\ ((b \wedge c') \vee (b' \wedge c)) \wedge \\ ((c \wedge a') \vee (c' \wedge a)) = 0 \end{aligned} \quad .$$

<u>Geometric formulation</u>:

The three points of intersection of opposite sides of a hexagon inscribed in a conic are collinear.

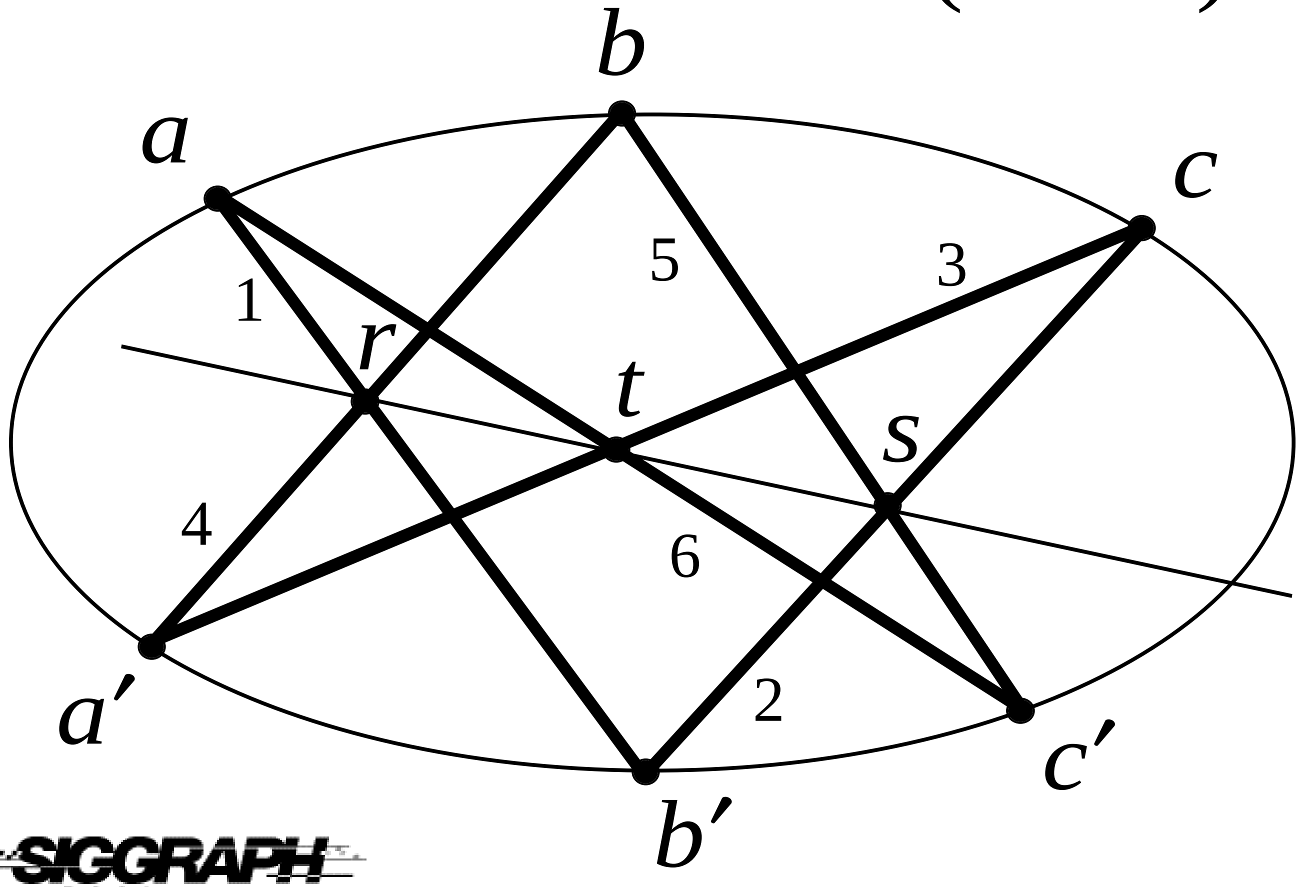
This is a property of the ***hexagrammum mysticum***, which Blaise Pascal discovered in 1640, at the age of 16.

Pascal's theorem (cont.)

$$r = (a \wedge b') \vee (a' \wedge b) = 1 \vee 4$$

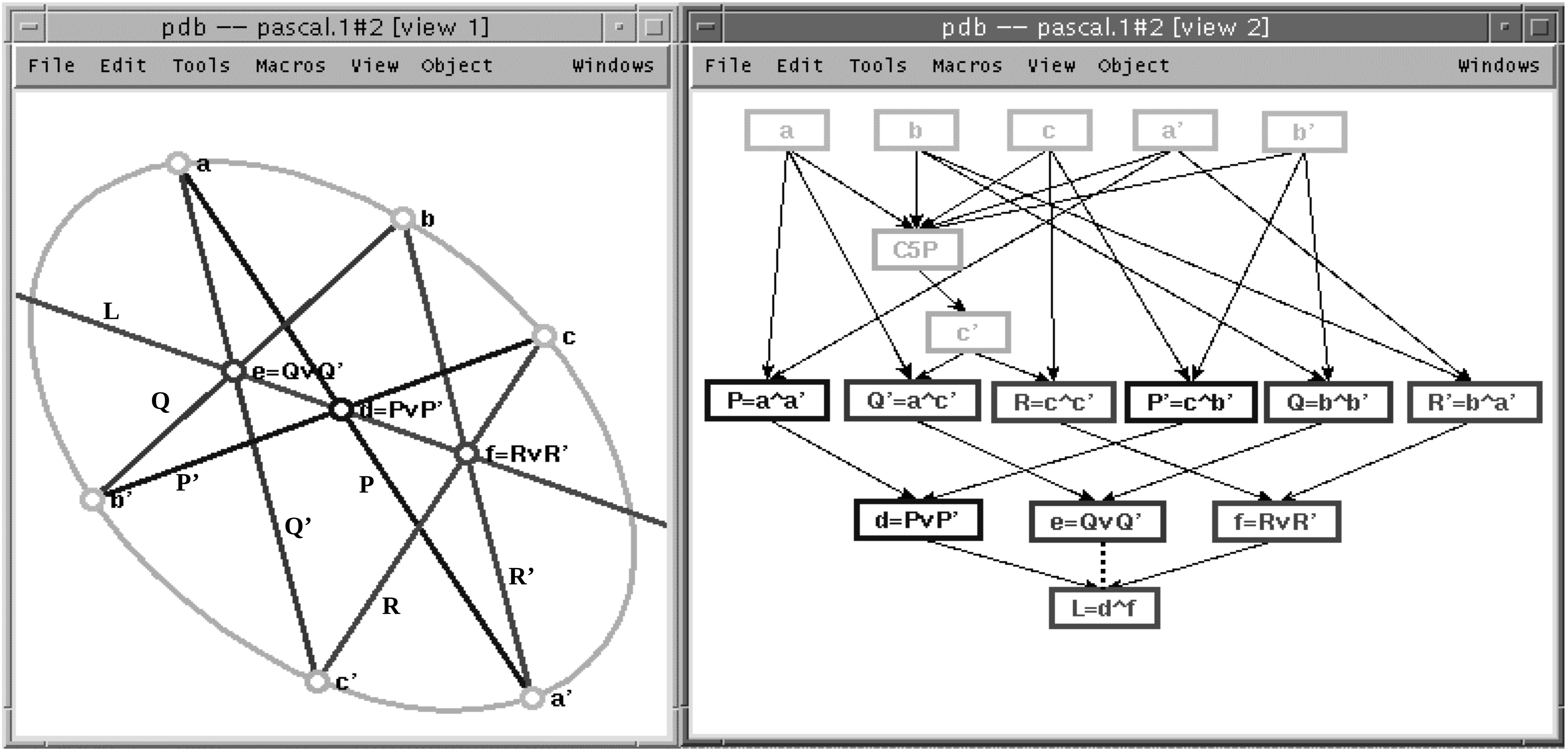
$$s = (b \wedge c') \vee (b' \wedge c) = 2 \vee 5$$

$$t = (c \wedge a') \vee (c' \wedge a) = 3 \vee 6$$



$$r \wedge s \wedge t = 0$$

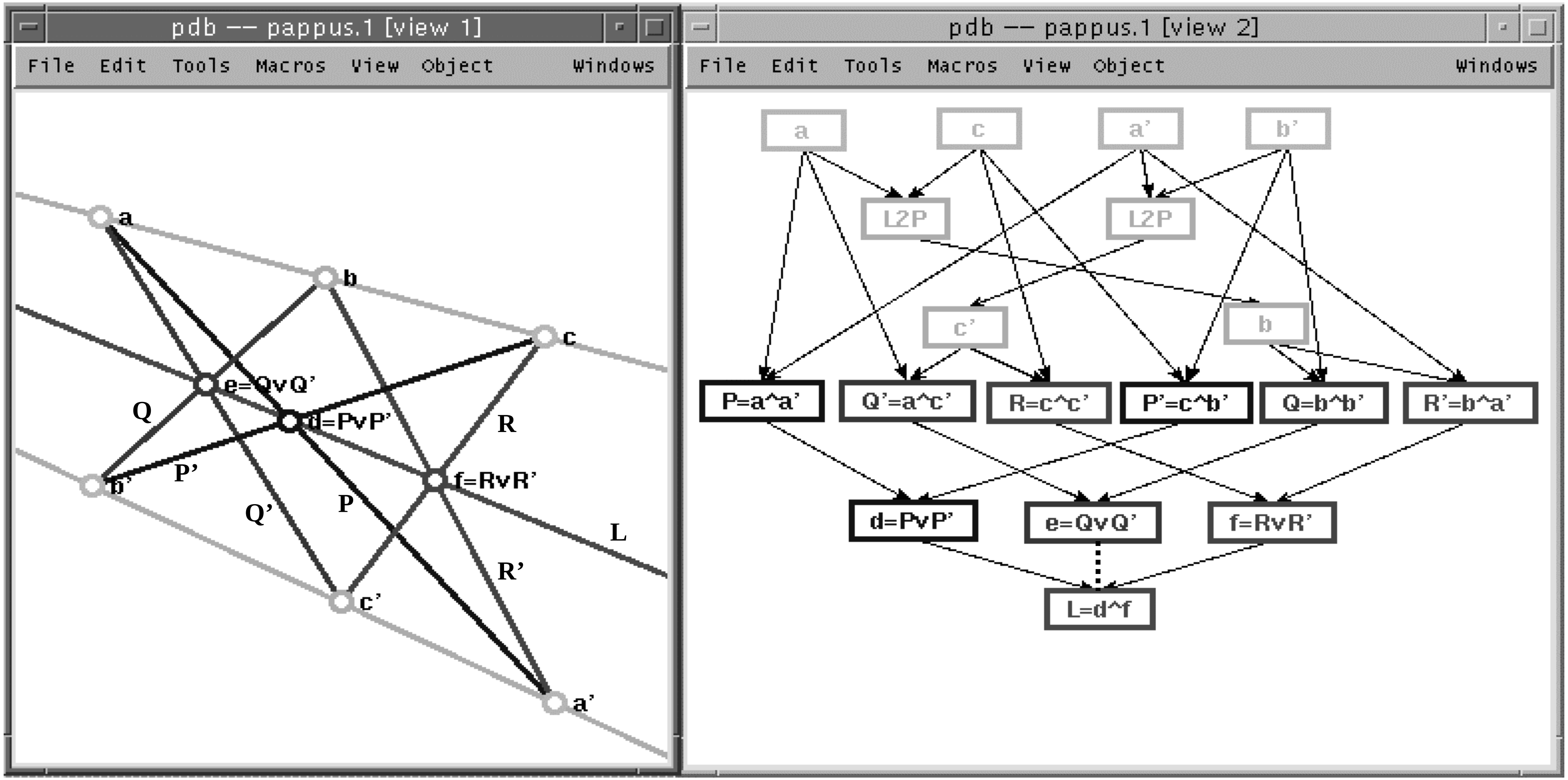
Pascal's theorem (cont.)



The three points of intersection of opposite sides of a hexagon inscribed in a conic are collinear.

SIGGRAPH 2001

Pappus' theorem (ca 350 A.D.)



If the conic degenerates into two straight lines,
Pappus' theorem emerges as a special case of Pascal's.

SIGGRAPH 2001 CREATE INTERACTION

Outermorphisms

Def: A mapping $f : \mathbf{G} \to \mathbf{G}$

is called an *outermorphism*

if

(i) f is linear.

(ii) $f(t) = t \quad \forall t \in \mathbf{R}$

(iii) $f(x \wedge y) = f(x) \wedge f(y) \quad \forall x, y \in \mathbf{G}$

(iv) $f(\mathbf{G}^k) \subset \mathbf{G}^k \quad \forall k \geq 0$

The induced outermorphism

Let $T : \mathbf{V} \rightarrow \mathbf{V}$ denote a linear mapping.

Fact: T induces an ***outermorphism*** $\underline{T} : \mathbf{G} \rightarrow \mathbf{G}$ given by

$$\underline{T}(a_1 \wedge \ldots \wedge a_k) = T(a_1) \wedge \ldots \wedge T(a_k)$$

$$\underline{T}(\lambda) = \lambda \quad , \quad \lambda \in \mathbf{R}$$

and linear extension.

Interpretation: $\underline{T}$ maps the *blades* of $\mathbf{V}$ in accordance with how T maps the *vectors* of $\mathbf{V}$.

Polarization with respect to a quadric in $\mathbf{P}^n$

Let $T : \mathbf{V}^{n+1} \rightarrow \mathbf{V}^{n+1}$ denote a *symmetric* linear map,

which means that $T(x) \cdot y = x \cdot T(y)$, $\forall x, y \in \mathbf{V}^{n+1}$.

The corresponding quadric (hyper)surface Q in $\mathbf{P}^n$

is given by $Q = \{x \in \mathbf{V}^{n+1} : x \cdot T(x) = 0 \ , \ x \neq 0\}$.

<u>Def</u>: The *polar* of the *k*-blade A with respect to Q

is the (n+1-k)-blade defined by

$$\mathrm{Pol}_Q(A) \equiv \underline{T}(A)^* \equiv \underline{T}(A) I^{-1}.$$

SIGGRAPH 2001 CREATE INTERACTION

Polarization (cont.)

Note: $T = id \Rightarrow \quad Q = \{x \in \mathbf{V}^{n+1} : x \cdot x = 0 \ , \ x \neq 0\}$.

In this case $\quad \mathrm{Pol}_Q(A) \equiv AI^{-1} \equiv A^*$

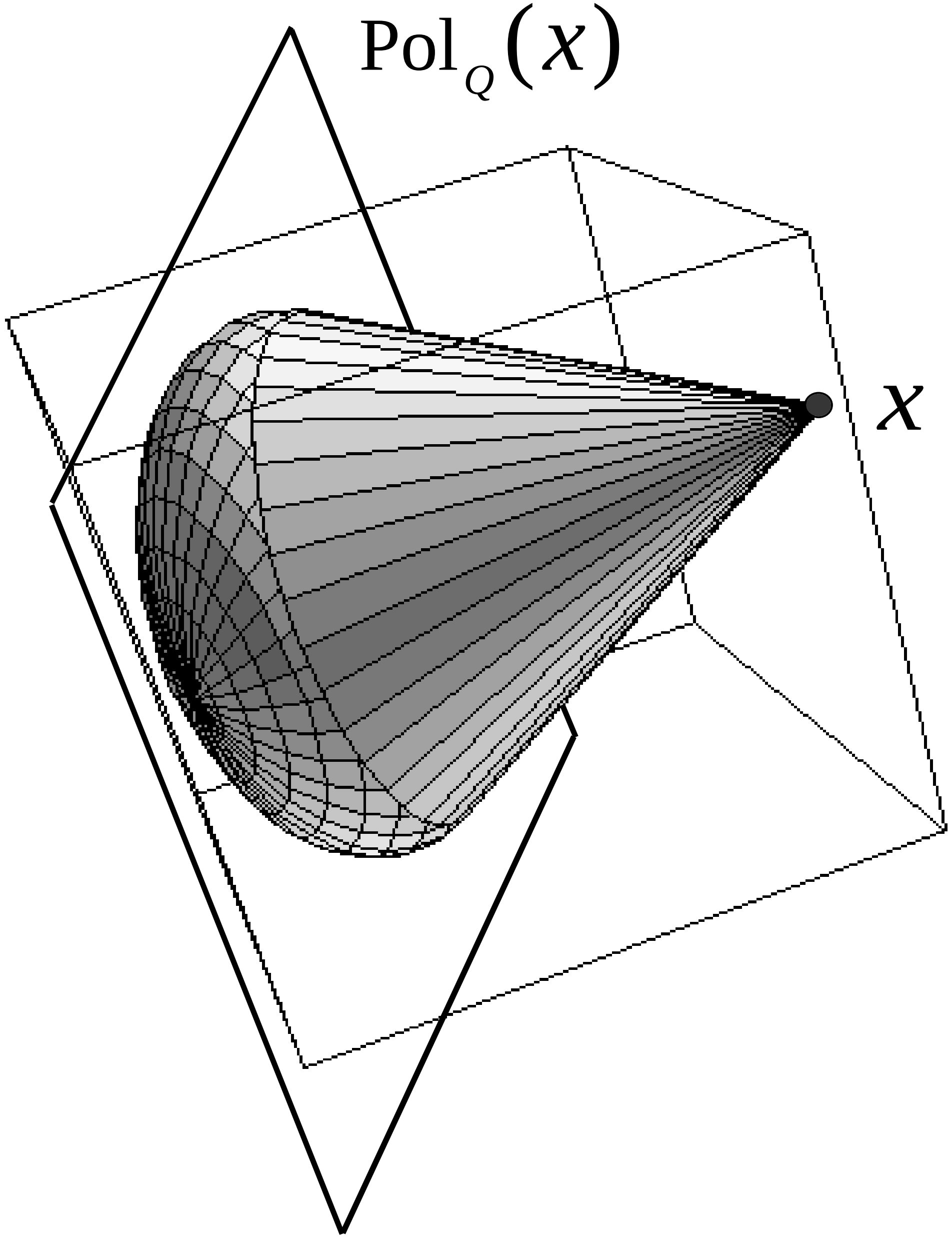
and polarization becomes identical to dualization.

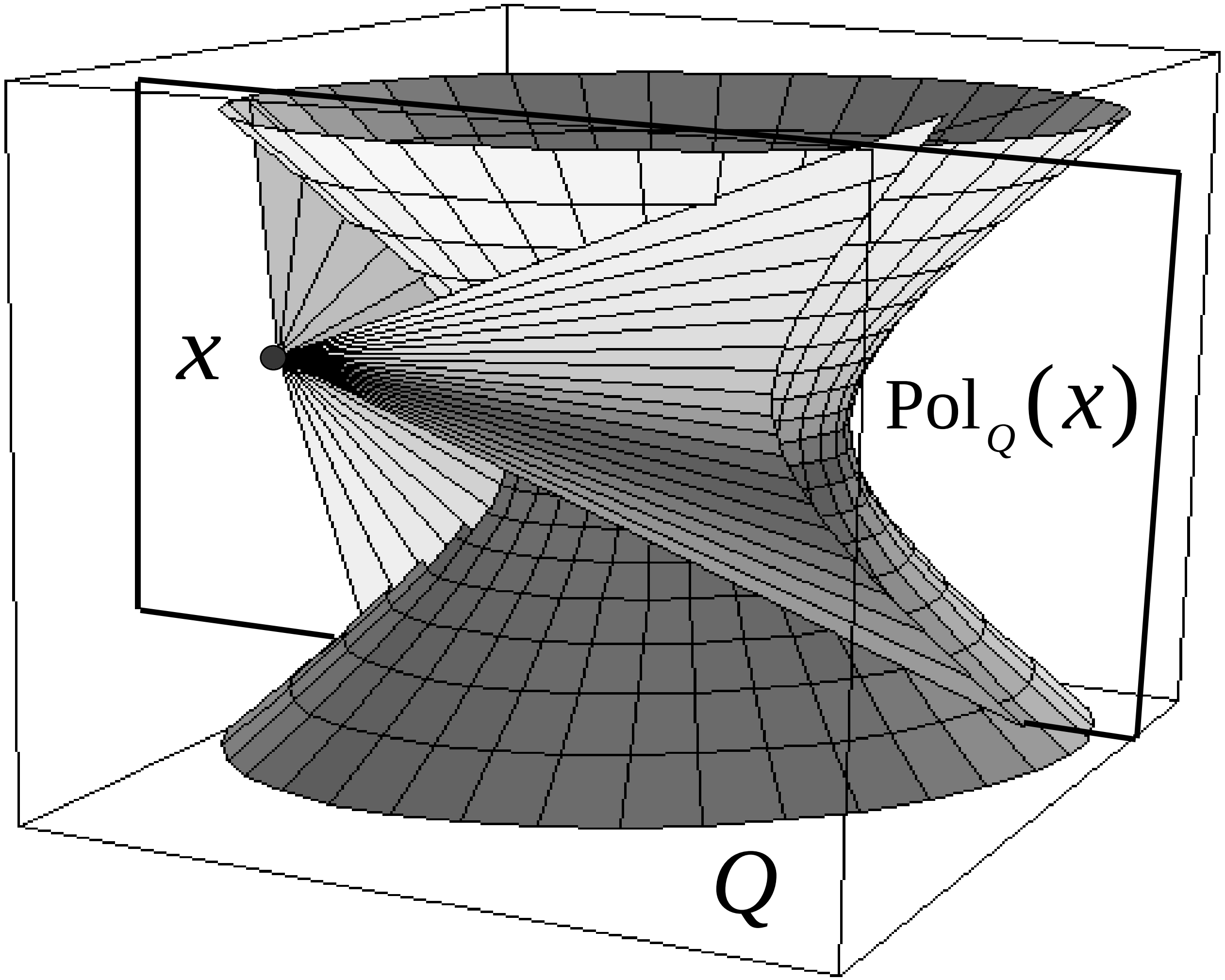
Fact: For a blade A we have

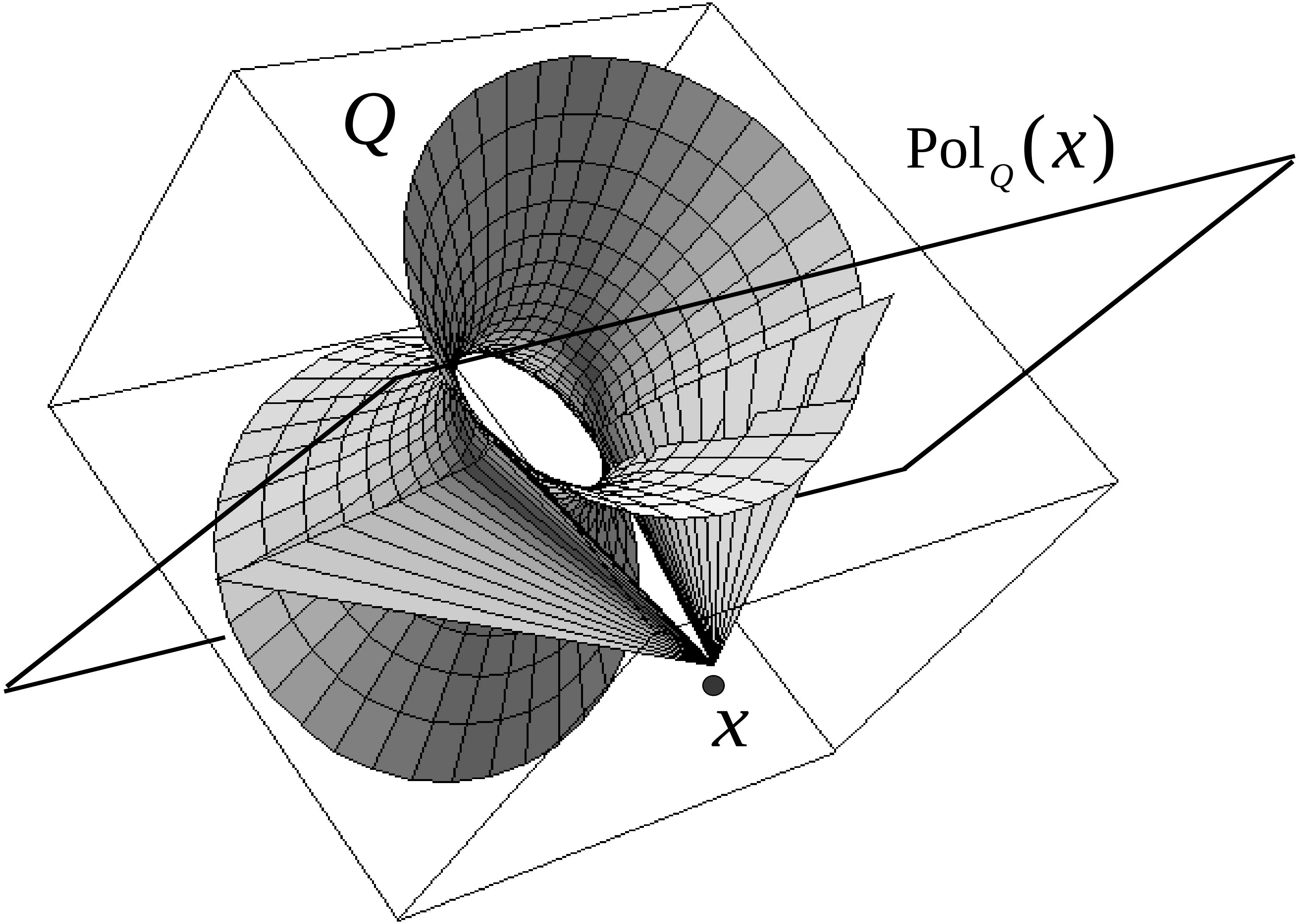
(i) $\quad \mathrm{Pol}_Q\left(\mathrm{Pol}_Q(A)\right) \doteq A$

(ii) If A is tangent to Q
then $\mathrm{Pol}_Q(A)$ is tangent to Q .

Especially: If x is a point on Q,
then $\mathrm{Pol}_Q(x)$ is the hyperplane
which is tangent to Q at the point x.

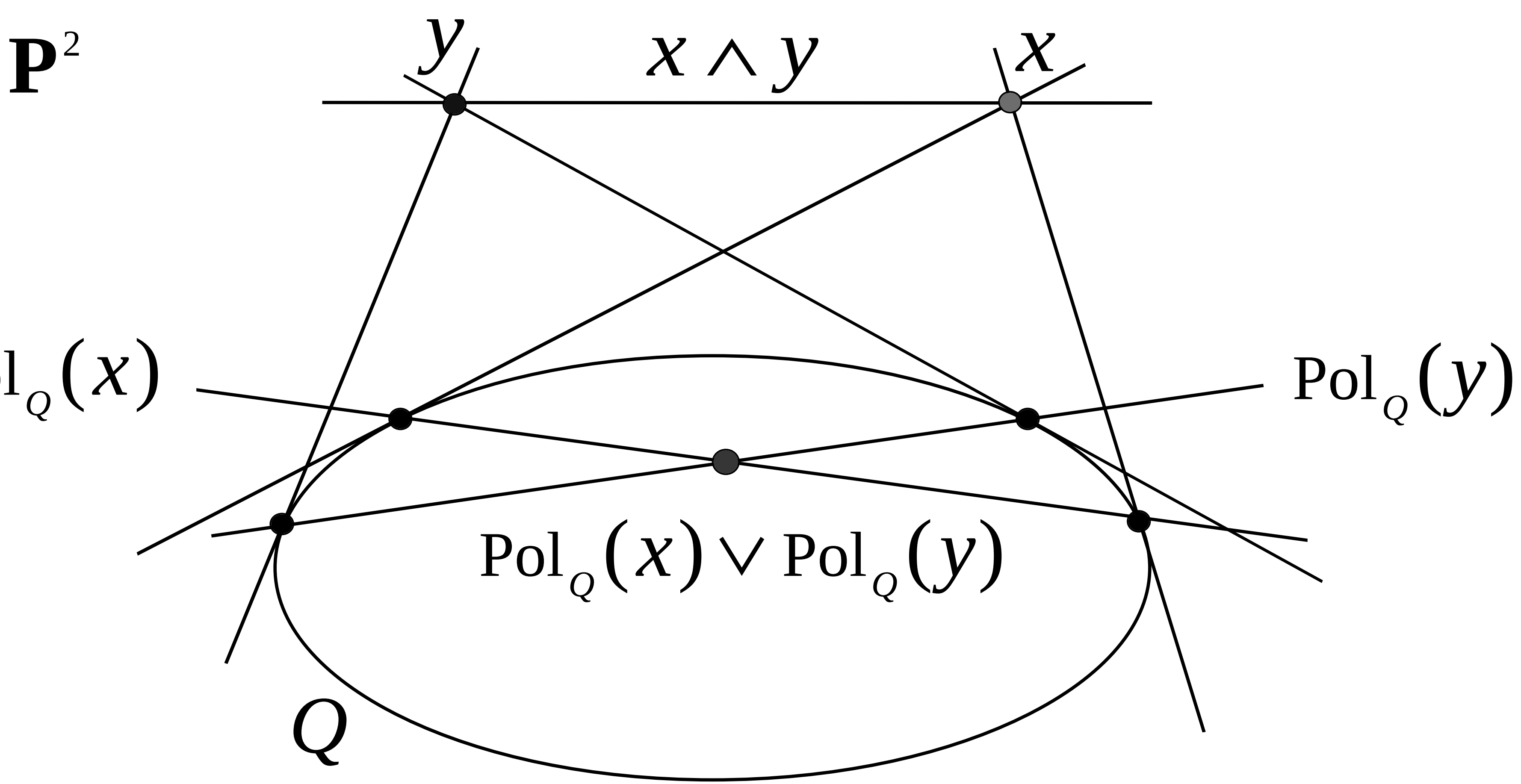

$\mathrm{Pol}_Q(x)$
x


x
$\mathrm{Pol}_Q(x)$
Q



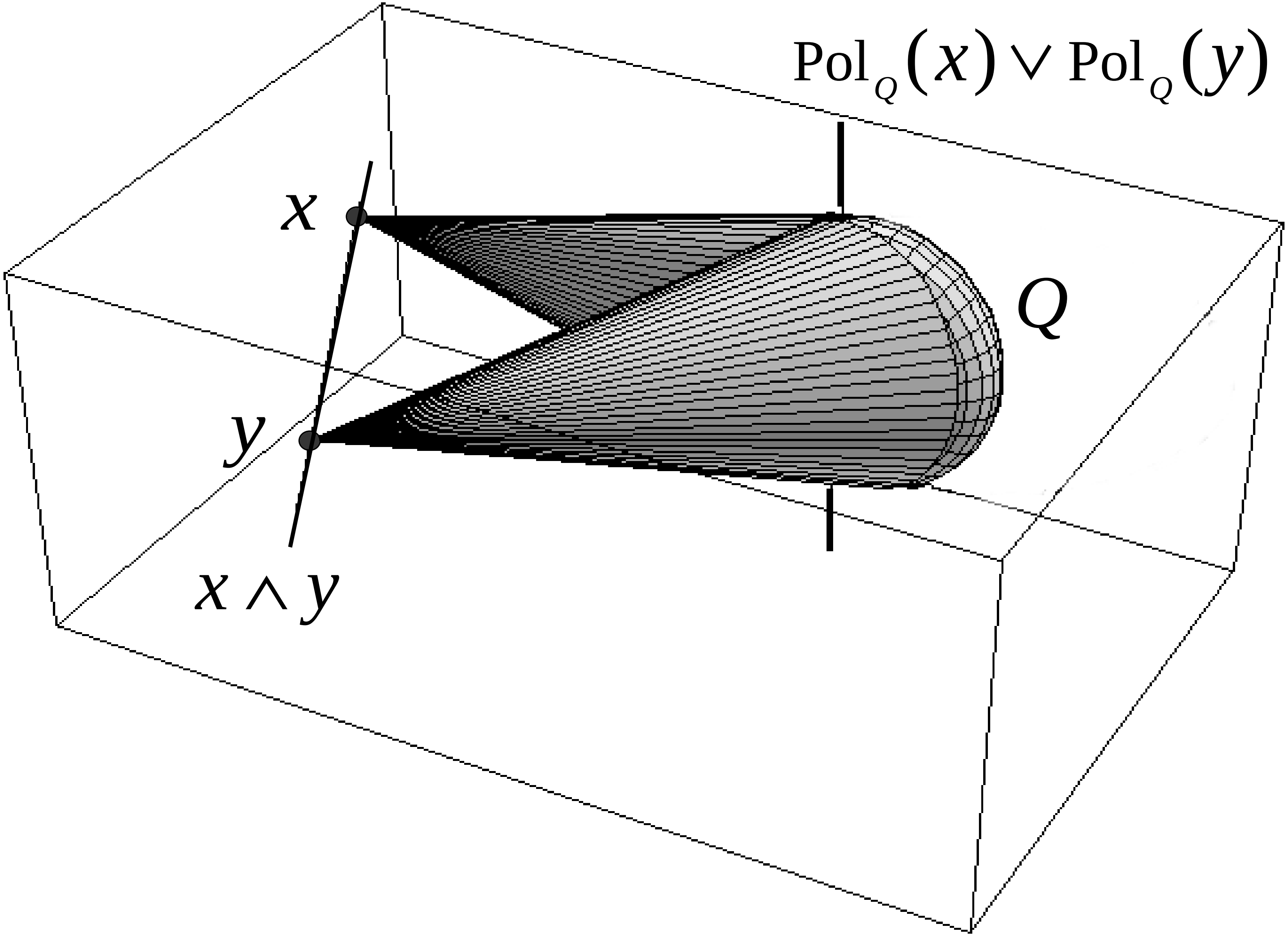
Q
$\mathrm{Pol}_Q(x)$
x

Polarity with respect to a conic



The polar of the ***join*** of x and y = $\mathrm{Pol}_Q(x \wedge y)$

is the ***meet*** of the polars of x and y = $\mathrm{Pol}_Q(x) \vee \mathrm{Pol}_Q(y)$


$\mathrm{Pol}_Q(x) \vee \mathrm{Pol}_Q(y)$
x
Q
y
$x \wedge y$

Polar reciprocity

Let $x, y \in \mathbf{V}^{n+1}$ represent two points in $\mathbf{P}^n$.

Then we have from the symmetry of T :

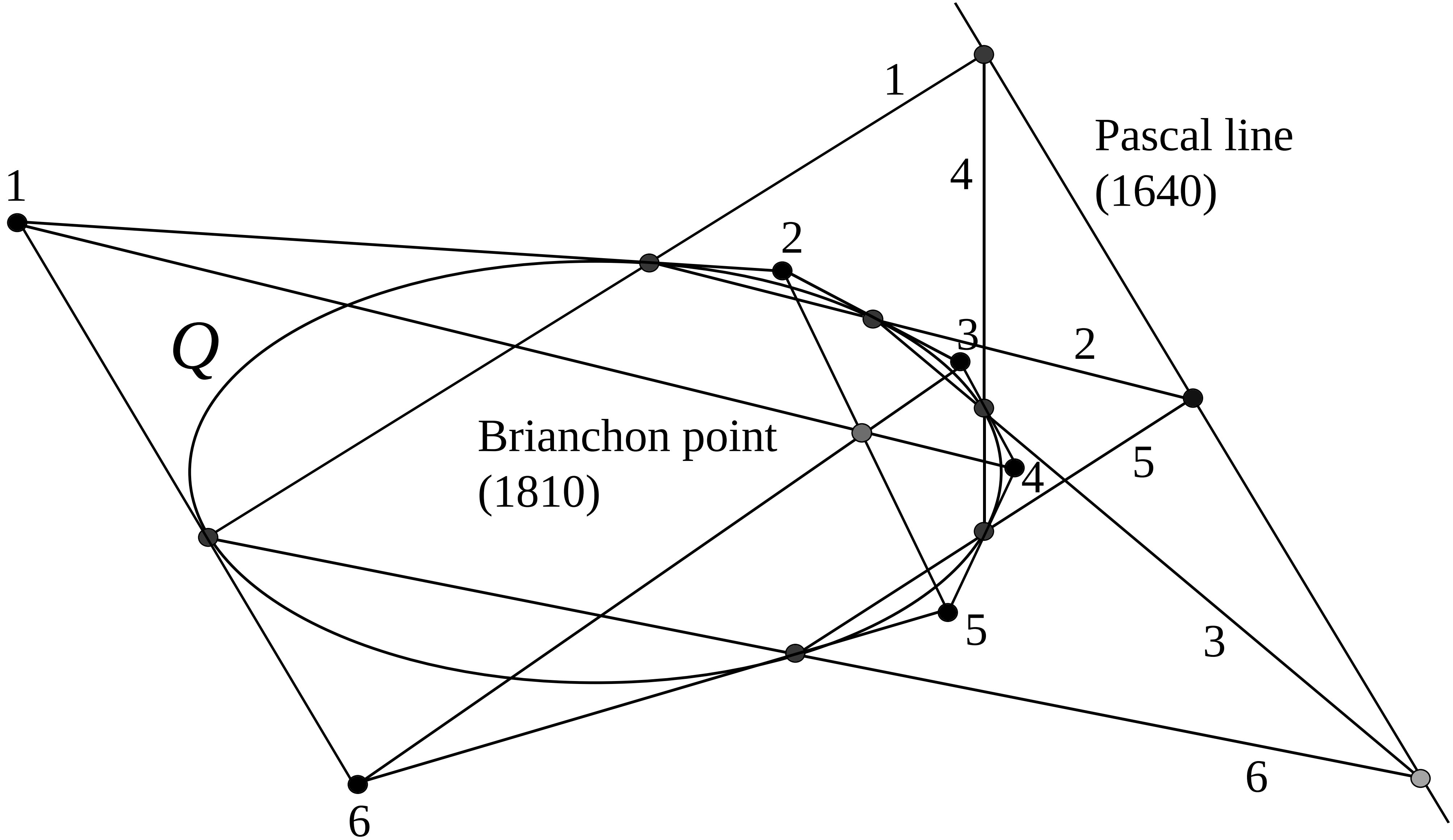
$$
\begin{aligned}
y \wedge T(x)^* &= y \wedge (T(x) I^{-1}) \\
&= (y \cdot T(x)) I^{-1} = (x \cdot T(y)) I^{-1} \\
&= x \wedge (T(y) I^{-1}) = x \wedge T(y)^*.
\end{aligned}
$$

Hence: $y \wedge \mathrm{Pol}_Q(x) = 0 \quad \Leftrightarrow \quad x \wedge \mathrm{Pol}_Q(y) = 0$

i.e. the point y lies on the polar of the point x if and only if x lies on the polar of y.

SIGGRAPH 2001 CREATE INTERACTION

Briánchon's theorem



The dual map

Let $f : \mathbf{G} \to \mathbf{G}$ be linear,

and assume that $I^2 \neq 0$.

Def: The dual map $\tilde{f} : \mathbf{G} \to \mathbf{G}$

is the linear map given by

$$\tilde{f}(x) = f(xI)I^{-1}$$

$$\begin{tikzcd}
\mathbf{G} \arrow[r, "f"] \arrow[d, "*"'] & \mathbf{G} \arrow[d, "*"] \\
\mathbf{G} \arrow[r, "\tilde{f}", "\tilde{f}(\tilde{x}) = \widetilde{f(x)}"'] \arrow[d, "*"'] & \mathbf{G} \arrow[d, "*"] \\
\mathbf{G} \arrow[r, "\tilde{\tilde{f}} = f"] & \mathbf{G}
\end{tikzcd}$$

Note:
$$f(x) = \tilde{f}(xI^{-1})I \quad = \tilde{f}(xI^{-1})I^2 I^{-2} I$$
$$= \tilde{f}(xI^{-1}I^2)I^{-1} \quad = \tilde{f}(xI)I^{-1} \quad = \tilde{\tilde{f}}(x)$$

Polarizing a quadric with respect to another

Let $S : \mathbf{G} \to \mathbf{G}$ and $T : \mathbf{G} \to \mathbf{G}$

be symmetric outermorphisms, and let

$$P = \{x \in \mathbf{G} : x * S(x) = 0 \ , \ x \neq 0\} \quad ,$$
$$Q = \{x \in \mathbf{G} : x * T(x) = 0 \ , \ x \neq 0\}$$

be the corresponding two quadrics.

Fact: Polarizing the multivectors of the quadric P
with respect to the quadric Q
gives a quadric R
with equation $x * (T \circ \tilde{S} \circ T(x)) = 0$
and we have $x \in P \iff \mathrm{Pol}_Q(x) \in R$.

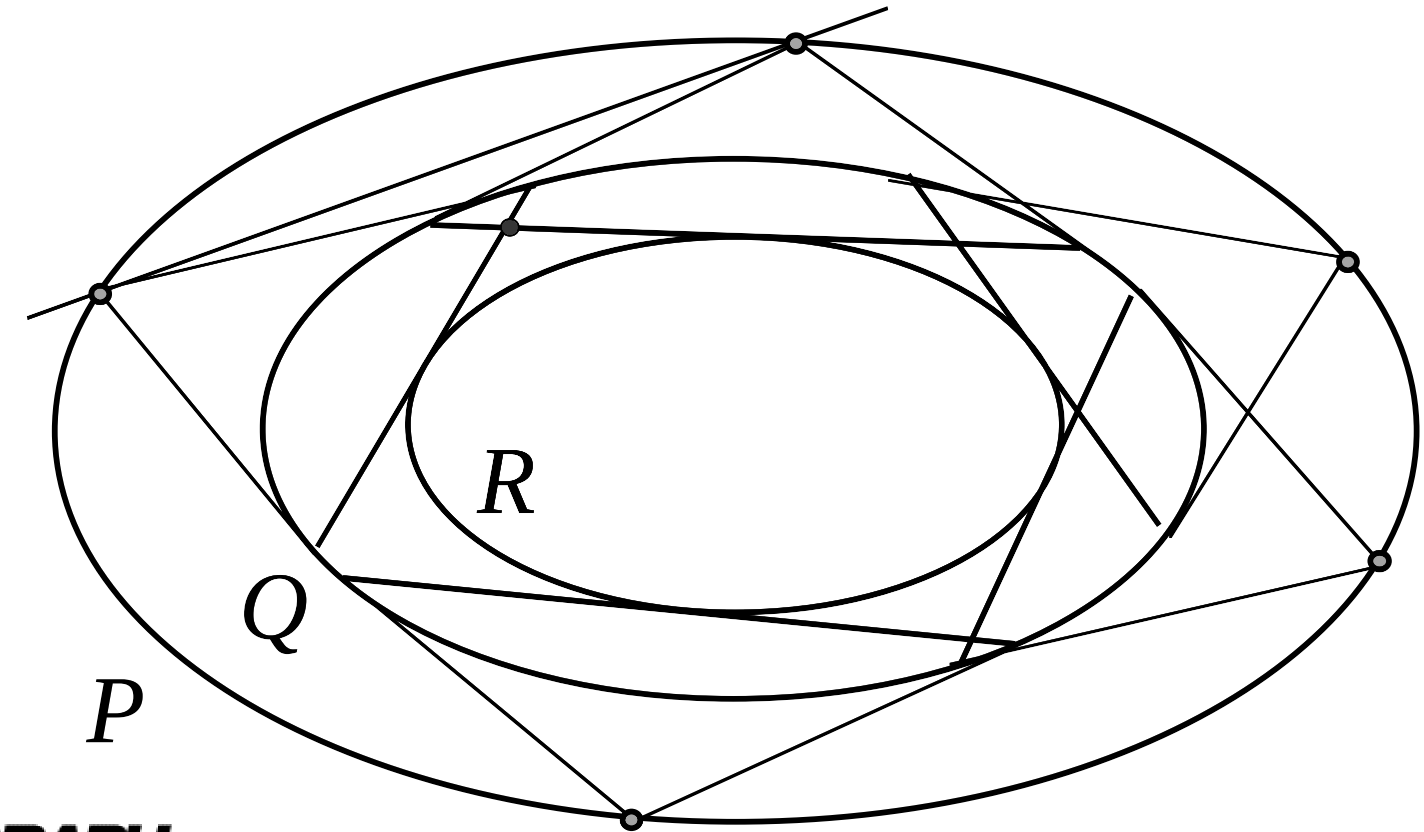
SIGGRAPH 2001 CREATE INTERACTION

The reciprocal quadric

Polarizing the quadric $P: \quad x * S(x) = 0$

with respect to the quadric $Q: \quad x * T(x) = 0$

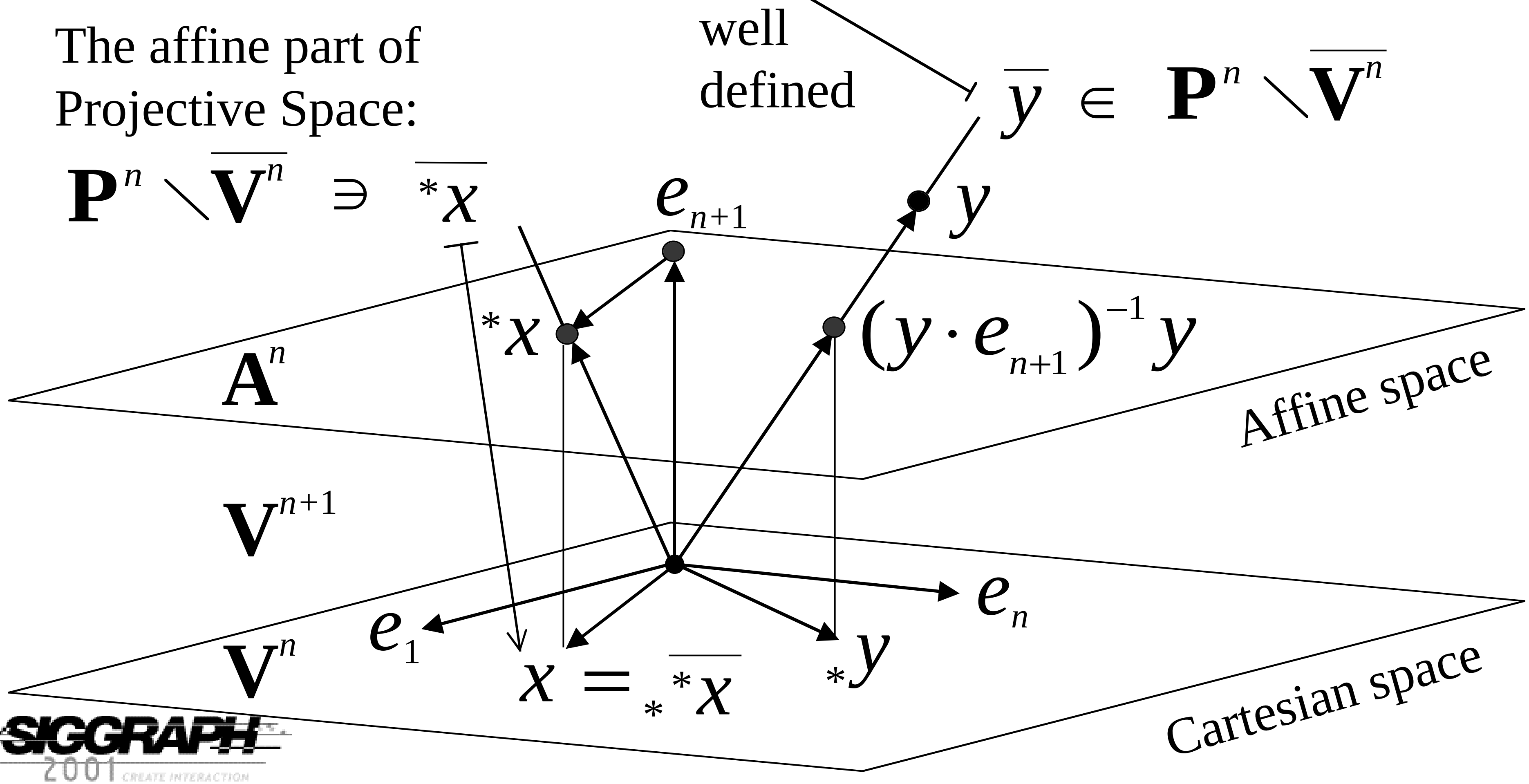
generates the quadric $R: \quad x * (T \circ \tilde{S} \circ T(x)) = 0$



Cartesian-Affine-Projective relationships

$$\mathbf{V}^n \ni x \longmapsto {}^*x = x + e_{n+1} \in \mathbf{A}^n$$

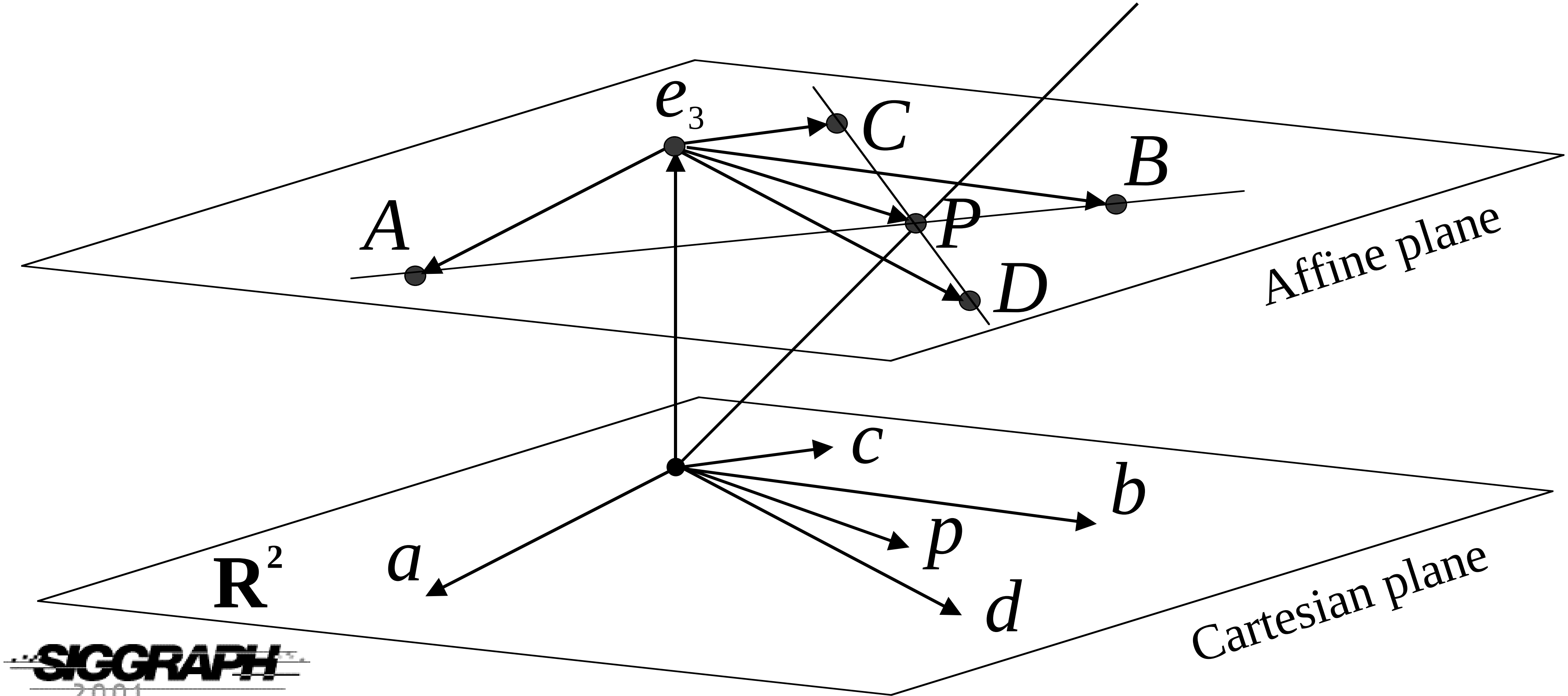
$$\mathbf{V}^{n+1} \setminus \mathbf{V}^n \ni y \longmapsto {}_*y = (y \cdot e_{n+1})^{-1} y - e_{n+1} \in \mathbf{V}^n$$



The intersection of two lines in the plane

$$a,b,c,d \xrightarrow{\;*(\;)\;} A,B,C,D$$

$$p \xleftarrow{\;{}_*(\;)\;} \bar{P} = (A \wedge B) \vee (C \wedge D)$$



SIGGRAPH 2001 CREATE INTERACTION

The intersection of two lines (cont.)

$$\bar{P} = (A \wedge B) \vee (C \wedge D)$$

$$= [A \wedge B \wedge C\ ]D - [A \wedge B \wedge D\ ]C \qquad = \alpha D - \beta C$$

$$[A \wedge B \wedge C\ ] = (A \wedge B \wedge C) I^{-1}$$

$$= \big((a + e_3) \wedge (b + e_3) \wedge (c + e_3)\big) e_3 e_2 e_1$$

$$= \big((a - c) \wedge (b - c) \wedge (c + e_3)\big) e_3 e_2 e_1$$

0 =

$$= \boxed{\big((a - c) \wedge (b - c) \wedge c\big)} e_3 e_2 e_1 +$$

$$\big((a - c) \wedge (b - c) \wedge e_3\big) e_3 e_2 e_1$$

$$= \big((a - c) \wedge (b - c) e_3\big) e_3 e_2 e_1$$

does not contain e_3

$$= \boxed{(a - c) \wedge (b - c) e_2 e_1} \quad \equiv \alpha$$

The intersection of two lines (cont.)

In the same way we get

does not contain e_3

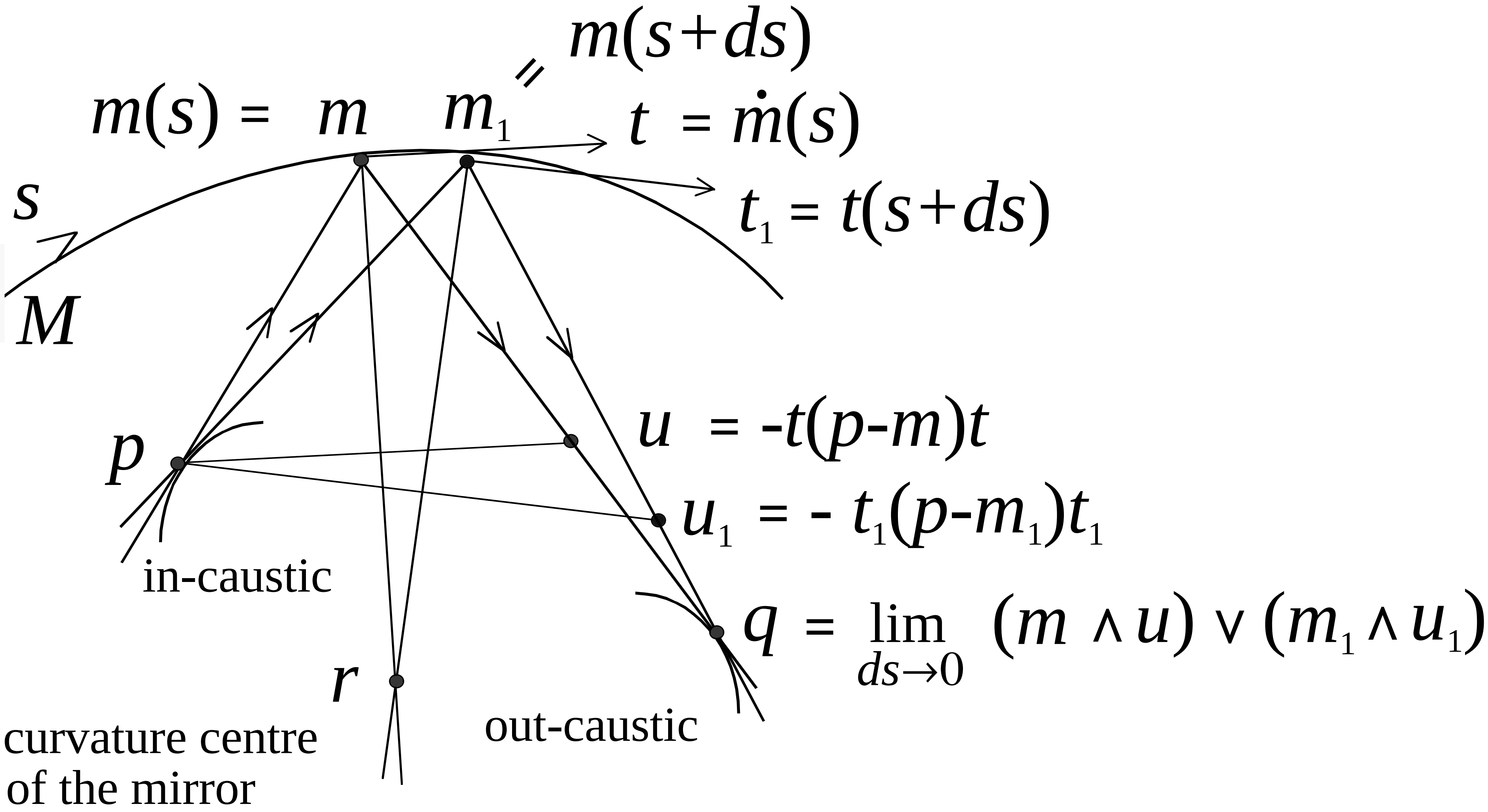
$$\beta = [A \wedge B \wedge D] = (A \wedge B \wedge D)I^{-1} = \boxed{(a-d) \wedge (b-d)e_2 e_1}$$

Hence we can write p as:

$$p = {}_*\overline{P} = {}_*(\alpha D - \beta C)$$
$$= ((\alpha D - \beta C) \cdot e_3)^{-1}(\alpha D - \beta C) - e_3$$
$$= (\alpha - \beta)^{-1}(\alpha d + \alpha e_3 - \beta c - \beta e_3) - e_3$$
$$= \boxed{(\alpha - \beta)^{-1}(\alpha d - \beta c)}$$

does not contain e_3

Reflection in a plane-curve mirror



Reflection in a plane-curve mirror (cont.)

Making use of the intersection formula deduced earlier

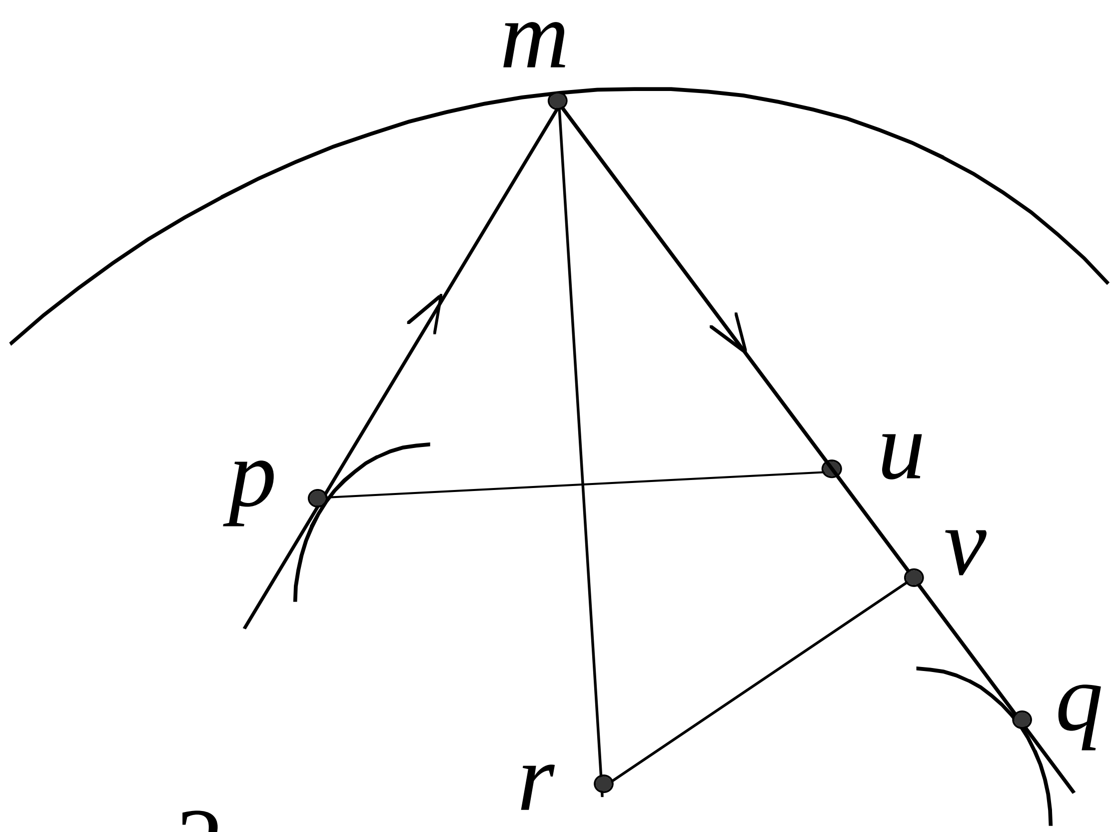
and introducing $n = \frac{\dot{t}}{|\dot{t}|}$ for the unit mirror normal

we get

$$q - m = \frac{((p-m)\cdot t)t - (p-m)\cdot n)n}{1 - 2|\ddot{m}|\frac{(p-m)^2}{(p-m)\cdot n}}$$

This is an expression of Tschirnhausen’s reflection law.

Reflection in a plane-curve mirror (cont.)



Tschirnhausens reflection formula

$$\frac{1}{|u-m|} \pm \frac{1}{|q-m|} = \frac{2}{|v-m|}$$

References:

Hestenes, D. & Ziegler, R., *Projective Geometry with Clifford Algebra*, Acta Applicandae Mathematicae 23, pp. 25-63, 1991.

Naeve, A. & Svensson, L., *Geo-Metric-Affine-Projective Unification*, Sommer (ed.), *Geometric Computations with Clifford Algebra*, Chapter 5, pp.99-119, Springer, 2000.

Winroth, H., *Dynamic Projective Geometry*, TRITA-NA-99/01, ISSN 0348-2953, ISRN KTH/NA/R--99/01--SE, Dissertation, The Computational Vision and Active Perception Laboratory, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, March 1999.

A Homogeneous Framework for Computational Geometry and Mechanics

David Hestenes
Department of Physics and Astronomy
Arizona State University, Tempe, Arizona, USA

Need: Euclidean geometry supplies essential mathematical underpinnings for physics, engineering and Computer-Aided Geometric Design.

Question: How should we formulate Euclidean Geometry to

- *facilitate* geometric modeling and analysis
- *optimize* computational efficiency?

A fundamental problem in the *Design of Mathematics*

GEOMETRIC ALGEBRA Website:

http://modelingnts.la.asu.edu

Can access related websites from this one

Standard model for Euclidean 3-space:

$$\mathcal{E}^3 \cong R^3 = \{\mathbf{x}\} = \text{3d real vector space}$$

Rigid body *motions and symmetries* described by

Euclidean group E(3)

$$\mathbf{x} \longrightarrow \mathbf{x}' = \underline{R}\,\mathbf{x} + \mathbf{a}$$

Invariant: $(\mathbf{x} - \mathbf{y})^2 = (\mathbf{x}' - \mathbf{y}')^2$

Computations: Require coordinates and Matrices

$$[\mathbf{x}'] = [\underline{R}]\,[\mathbf{x}] + [\mathbf{a}]$$

Drawbacks:

Extraneous information

(from arbitrary choice of coordinates)

Redundant matrix elements

(9 elements for 3 degrees of freedom)

Difficult to interpret

(parameters mixed in matrix elements)

Computationally inefficient

(9×9 vs. 3×3 multiplications)

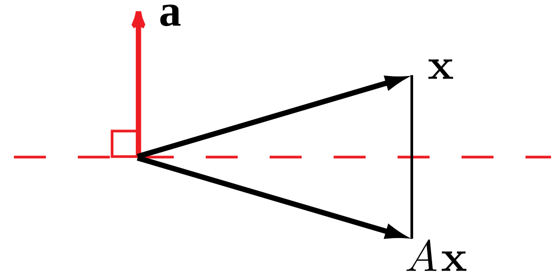
Geometric algebra eliminates need for coordinates:

Canonical form for orthogonal transformations:

$$\underline{R}\,\mathbf{x} = \varepsilon_R R\,\mathbf{x}\,R^{-1} \qquad \text{parity } \varepsilon_R = \pm 1$$

Reflection represented by a vector $\mathbf{a}$

$$\underline{A}\,\mathbf{x} = -\mathbf{a}\mathbf{x}\mathbf{a}^{-1}$$

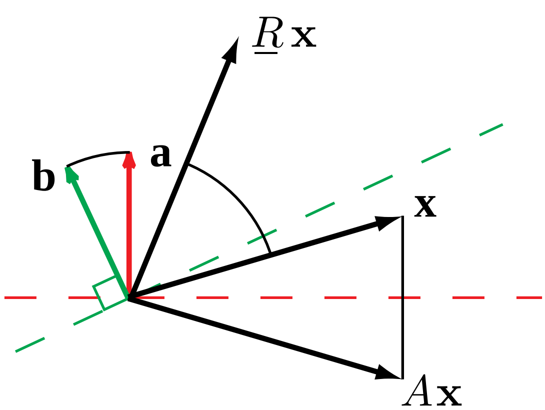


Rotation = double reflection

$$\underline{R}\,\mathbf{x} = \underline{B}\,\underline{A}\,\mathbf{x}$$

$$= -\mathbf{b}(-\mathbf{a}\mathbf{x}\mathbf{a}^{-1})\,\mathbf{b}^{-1}$$

$$= (\mathbf{b}\mathbf{a})\,\mathbf{x}\,(\mathbf{a}^{-1}\mathbf{b}^{-1}) = R\mathbf{x}R^{-1}$$



Rotation represented by a versor (rotor, spinor, *quaternion*)

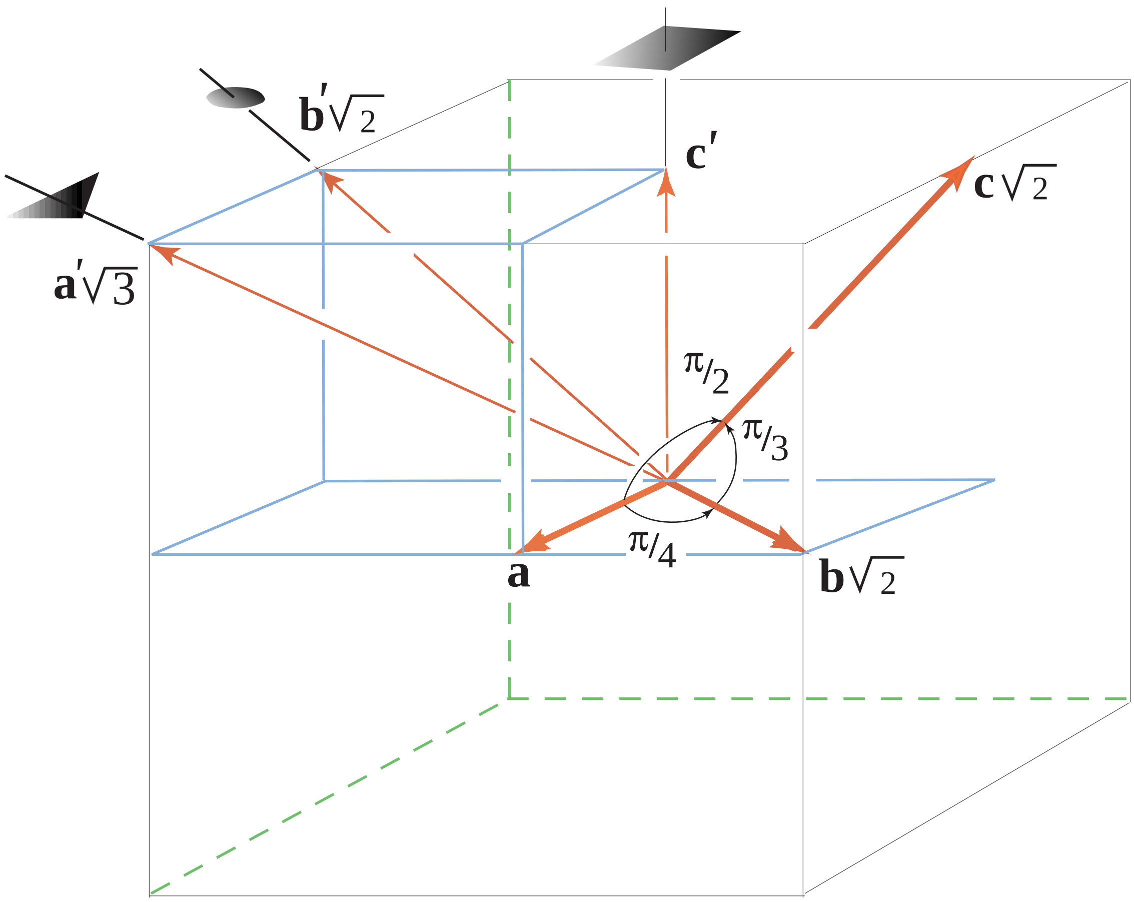
$$R = \mathbf{b}\mathbf{a}$$

Reduces composition of rotations/reflections to geometric product of versors:

$$\underline{R}_2\underline{R}_1 = \underline{R}_3 \qquad \leftrightarrow \qquad R_2 R_1 = R_3$$

Extensive treatment in (Hestenes 1986)

Symmetries of the Cube



3 Generators: $\mathbf{a},\ \mathbf{b},\ \mathbf{c}$ $\qquad \mathbf{a}^2 = \mathbf{b}^2 = \mathbf{c}^2 = 1$

Relations:
$$(\mathbf{ab})^4 = -1$$
$$(\mathbf{bc})^3 = -1$$
$$(\mathbf{ca})^2 = -1$$

Problem:

Vector space model for $\mathcal{E}^3 \cong R^3 = \{\mathbf{x}\}$

singles out one point for special treatment

as the origin $\mathbf{x} = \mathbf{0}$

Drawbacks:

Often need to shift origin to

- simplify calculations
- avoid dividing by zero
- prove results independent of origin

Rigid body displacements combine

- rotations multiplicatively and
- translations additively,
- destroying the simplicity of both

Solution:

Design a *homogeneous model* for $\mathcal{E}^3$ that

- treats all points equally
- treats both rotations and translations multiplicatively

[“homogeneous” in the sense of *homogeneous coordinates*]

Homogeneous Euclidean Geometry

Arena: *Minkowski Algebra* $\mathcal{R}_{n+1,1} = \mathcal{G}(\mathcal{R}^{n+1,1})$

$\mathcal{R}^{n+1,1}$ = *vector space* with signature $(n+1,1)$

Null cone: $\{x \,|\, x^2 = 0\}$

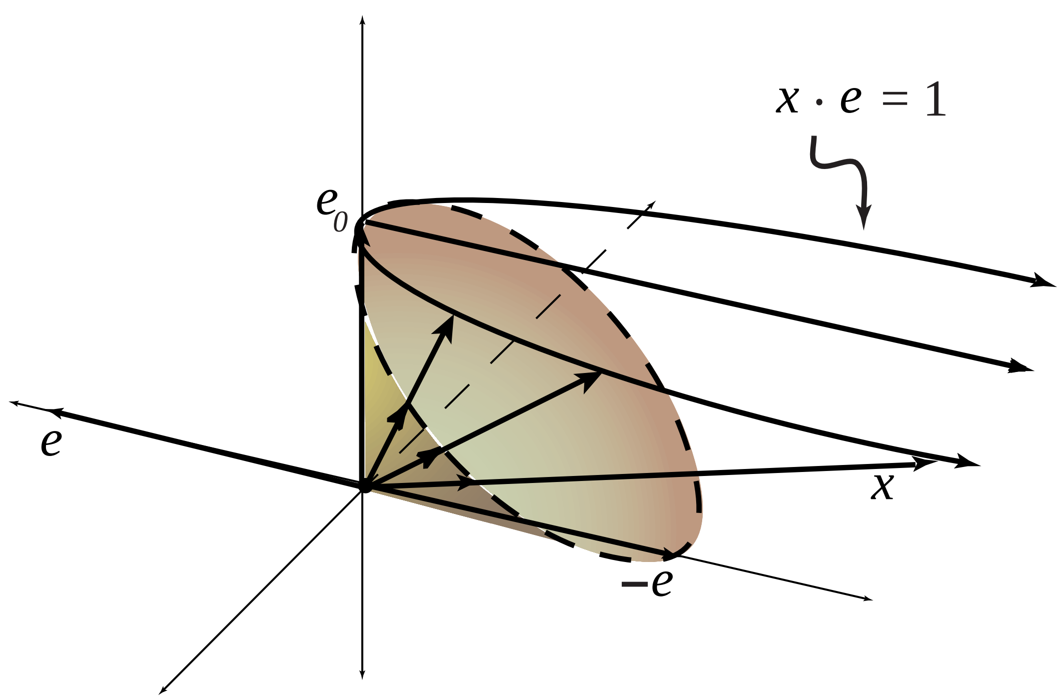
Hyperplane: $\{x \,|\, x \cdot e = 1\}$

Homogeneous model of Euclidean space:

$$\mathcal{E}^n \cong \{x \,|\, x^2 = 0, \;\; x \cdot e = 1\} \qquad\qquad e^2 = 0$$

x called a *point* in $\mathcal{E}^n$ $\qquad$ e = *point at infinity*

Horosphere (F. A. Wachter 1792-1817), (originally with coordinates)



Geometric algebra essential to make the horosphere

coordinate-free and *computationally efficient*

Hermann Günther Grassmann (1809-1877)

Laid the foundations for geometric algebra

Set a direction for future development

But he failed to reach his main goal,

ultimately concluding that it was impossible

Grassmann's Goal: To formulate

$$\boxed{\textit{Euclidean geometry as an algebra of geometric objects}}$$

(points, lines, planes, circles, etc.)

$$\left\{\begin{matrix}\text{Synthetic}\\ \text{geometry}\end{matrix}\right\} \xleftrightarrow[\text{with}]{\text{integrated}} \left\{\begin{matrix}\text{Computational}\\ \text{geometry}\end{matrix}\right\}$$

$$\text{Interpretation} \xleftrightarrow[\text{to}]{\text{married}} \text{Computation}$$

The homogeneous model for $\mathcal{E}^n$ within geometric algebra

reaches Grassmann's Goal with many surprises!!!

Euclidean Geometry is inherent in algebraic properties of homogeneous points

I. Point-to-point distance from inner products:

$$(x-y)^2 = \cancelto{0}{x^2} - 2x \bullet y + \cancelto{0}{y^2}$$

$$\boxed{(x-y)^2 = -2x \bullet y \ = \text{Euclidean distance}}$$

$\Longrightarrow$ Points *must* be null vectors! Why Grassmann failed!

$\Longrightarrow$ Pythagorean theorem, triangular inequality, etc.

II. Lines and Planes from outer products

$A = a \wedge b \wedge e = \textit{line segment}$ or *line* thru pts a and b

$$A^2 = (a \wedge b \wedge e)^2 = (a-b)^2 = -2a \bullet b = (\text{length})^2$$

$P = a \wedge b \wedge c \wedge e = \textit{plane segment}$ or *plane*

$$P^2 = (a \wedge b \wedge c \wedge e)^2 = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & a \bullet b & a \bullet c \\ 1 & b \bullet a & 0 & b \bullet c \\ 1 & c \bullet a & c \bullet b & 0 \end{vmatrix} = 4(\text{area})^2$$

Cayley-Menger determinant (another historical curiosity)

Cayley, 1841 (Volumes) $-Menger, 1931\,(distance\,geometry)$

Dress and Havel, 1993 (relation to GA)

III. Geometric relations from algebraic products

e.g.: Point x lies on the line A or plane P if and only if

$x \wedge A = 0$ $\qquad$ $x \wedge P = 0$

The homogeneous model of $\mathcal{E}^n$ is related to the standard inhomogeneous model by a

Conformal split:

A 1-to-1 mapping of $\mathcal{E}^n = \{x\}$

onto

the *pencil of lines* $\mathcal{R}^n = \{\mathbf{x}\}$ through a fixed point e_0,

defined by

$$x \longrightarrow \boxed{\mathbf{x} = x \wedge E} = x \wedge e_0 \wedge e$$

and the inverse mapping

$$\mathbf{x} \longrightarrow \boxed{x = \mathbf{x}E - \tfrac{1}{2}\mathbf{x}^2 e + e_0}$$

where $e_0 \wedge e = E \implies E^2 = 1$

and $eE = e = -Ee$ (absorption)

This implies the isomorphism: $\boxed{\mathcal{E}^n \cong \mathcal{R}^n}$

and determines a split of the whole algebra:

$$\boxed{\mathcal{R}_{n+1,1} = \mathcal{R}_n \otimes \mathcal{R}_{1,1}}$$

where $\mathcal{R}_n = \mathcal{G}(\mathcal{R}^n)$,

and $\mathcal{R}_{1,1}$ has the basis: $\{1, e, e_0, E = e_0 \wedge e\}$

Conformal Splits for Simplexes

Point: $a = (\mathbf{a} - \frac{1}{2}\,\mathbf{a}^2 e - e_0)\,E = E(\mathbf{a} + \frac{1}{2}\,\mathbf{a}^2 e + e_0)$

$$= \mathbf{a}E - \tfrac{1}{2}\,\mathbf{a}^2 e + e_0$$

Product: $ab = (aE)(Eb) = (\mathbf{a} + \frac{1}{2}\,\mathbf{a}^2 e + e_0)(\mathbf{b} - \frac{1}{2}\,\mathbf{b}^2 e - e_0)$

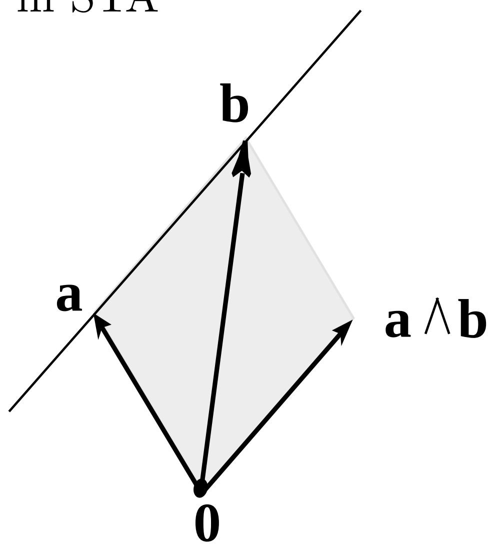
$$= \underbrace{-\tfrac{1}{2}(\mathbf{a}-\mathbf{b})^2}_{a\cdot b} + \underbrace{\mathbf{a}\wedge\mathbf{b} + \tfrac{1}{2}(\mathbf{a}^2\mathbf{b} - \mathbf{b}^2\mathbf{a})e + (\mathbf{b}-\mathbf{a})e_0 - \tfrac{1}{2}(\mathbf{b}^2 - \mathbf{a}^2)E}_{a\wedge b}$$

Similar to the mess from the spacetime split in STA

- *Line* (or *line segment*) thru a, b, e

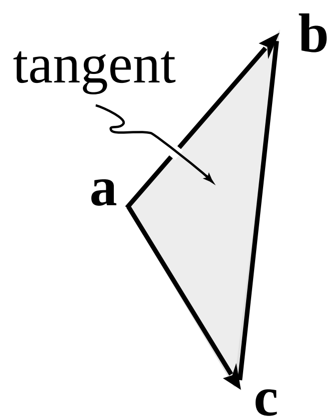
$$a\wedge b\wedge e = \underbrace{\mathbf{a}\wedge\mathbf{b}}_{\text{moment}}\,e + \underbrace{(\mathbf{b}-\mathbf{a})}_{\text{tangent}}E$$

Plücker coordinates



- *Plane* (or *plane segment*)

$$a\wedge b\wedge c\wedge e = \underbrace{\mathbf{a}\wedge\mathbf{b}\wedge\mathbf{c}}_{\text{moment}}\,e + \underbrace{(\mathbf{b}-\mathbf{a})\wedge(\mathbf{c}-\mathbf{a})}_{\text{tangent}}E$$



$$\mathbf{a}\wedge[(\mathbf{b}-\mathbf{a})\wedge(\mathbf{c}-\mathbf{a})] = \mathbf{a}\wedge\mathbf{b}\wedge\mathbf{c}$$

A Surprising algebra of spheres

All *spheres and hyperspheres in* $\mathcal{E}^n$ *are uniquely represented by "positive" vectors in* $\mathcal{R}^{n+1}$

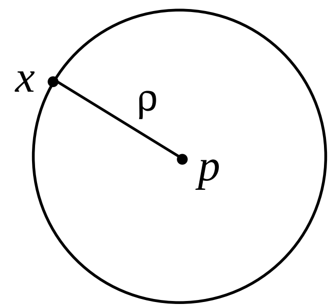
- $\mathcal{S}^{n-1}(\rho, p) =$ *sphere* with radius ρ and center p

  represented by vector s with

$$\frac{s^2}{(s \cdot e)^2} = \rho^2 \qquad p = \frac{s}{s \cdot e} - \tfrac{1}{2}\rho^2 e \quad \Longrightarrow \quad p^2 = 0$$

Can simplify with constraint $s \cdot e = 1$, so

$$s^2 = \rho^2 > 0, \quad p = s - \tfrac{1}{2}\rho^2 e$$



Eqn. for sphere: $x \cdot s = 0\,, \quad x^2 = 0$

Conformal split: $s = \mathbf{p}E + \frac{1}{2}(\rho^2 - \mathbf{p}^2)\,e + e_0$

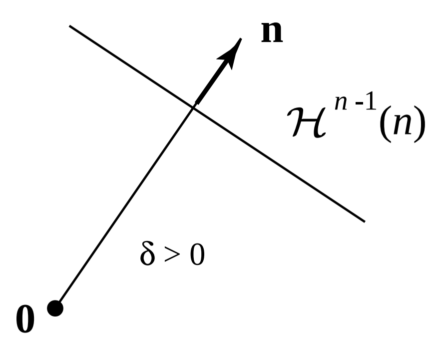
$$x \cdot p = -\tfrac{1}{2}\rho^2 \quad \Longrightarrow \quad \rho^2 = (\mathbf{x} - \mathbf{p})^2$$

- $\mathcal{H}^{n-1}(n) =$ *hyperplane* rep. by vector n with $\begin{cases} n \cdot e = 0 \\ n^2 = 1 \end{cases}$

Conformal split: $n = \mathbf{n}E - \delta e \quad \Longrightarrow \quad n^2 = \mathbf{n}^2 = 1$

Eqn. for hyperplane: $x \cdot n = 0\,, \; x^2 = 0$

or: $\qquad \mathbf{x} \cdot \mathbf{n} = \delta$



The homogeneous model of $\mathcal{E}^n$
maps all spheres and hyperplanes $\mathcal{R}^n$
into hyperplanes thru the origin in $\mathcal{R}^{n+1,1}$:

$$\{x \,|\, x \cdot s = 0\,,\; s^2 > 0\;,\;\; s \cdot e \geq 0\;,\;\; x^2 = 0;\;\; x, s \in \mathcal{R}^{n+1,1}\}$$

$x \cdot s = 0$ is the eqn for a hyperplane thru the origin

i.e., a $(n+1)$-dim subspace of $\mathcal{R}^{n+1,1}$

$s \cdot e = 0$ A sphere thru $e = \infty$ is a hyperplane

- Sphere determined by $n+1$ points: a_0, a_1, a_2, $\ldots$ a_n

$$\tilde{s} \equiv a_0 \wedge a_1 \wedge a_2 \wedge \cdots \wedge a_n \neq 0 \qquad \text{tangent form}$$

Radius ρ and center p given by duality:

$$s = -(a_0 \wedge a_1 \wedge a_2 \wedge \cdots \wedge a_n)I \qquad \text{normal form}$$

$$\left(\frac{s}{s \cdot e}\right)^2 = \rho^2\,, \qquad p = \frac{s}{s \cdot e} - \tfrac{1}{2}\rho^2 e$$

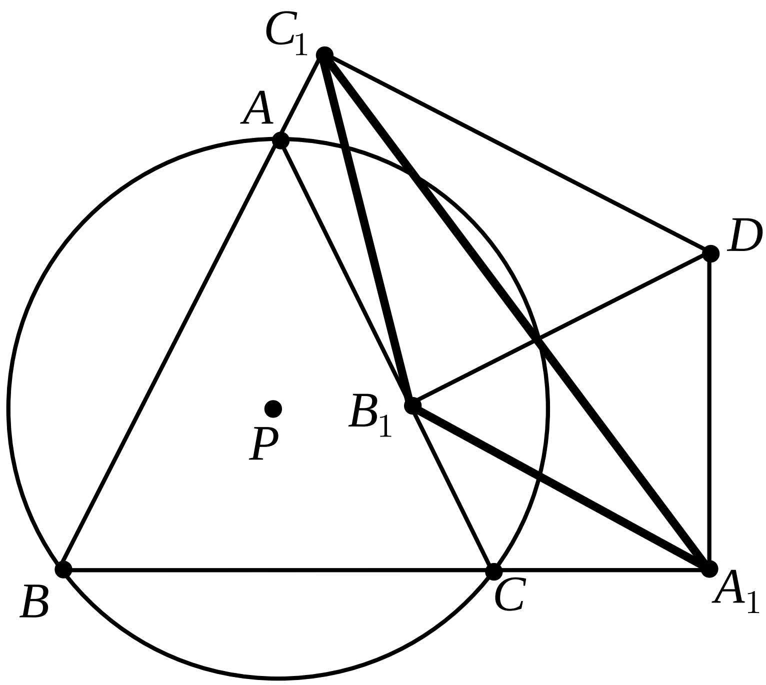
- Hyperplane = sphere thru ∞, say $a_0 = e$

$$\tilde{n} = e \wedge a_1 \wedge a_2 \wedge \cdots \wedge a_n \qquad \text{tangent form}$$

$$n = -(e \wedge a_1 \wedge a_2 \wedge \cdots \wedge a_n)I \qquad \text{normal form}$$

Eqn for sphere: $x \wedge \tilde{s} = 0 \iff x \cdot s = 0$

Example: Simson's construction



$\tilde{s} = A \wedge B \wedge C =$ circumcircle of triangle $e \wedge A \wedge B \wedge C$

$$\rho^2 = \left(\frac{s}{s \bullet e}\right)^2 = \frac{\tilde{s}^\dagger \tilde{s}}{(\tilde{s} \wedge e)^2} = \frac{(C \wedge B \wedge A) \bullet (A \wedge B \wedge C)}{(e \wedge A \wedge B \wedge C)^2}$$

Identity: $$\boxed{e \wedge A_1 \wedge B_1 \wedge C_1 = \frac{A \wedge B \wedge C \wedge D}{2\rho^2}}$$ (H. Li)

$$A \wedge B \wedge C \wedge D = 0 \quad \Longleftrightarrow \quad e \wedge A_1 \wedge B_1 \wedge C_1 = 0$$

D lies on circumcircle $\Longleftrightarrow$ A_1, B_1, C_1 are collinear

Rigid Displacements and Symmetries

$$\left\{\begin{matrix}\text{Lorentz group} \\ \text{on } \mathcal{R}^{n+1,1}\end{matrix}\right\} \cong \left\{\begin{matrix}\text{Conformal group} \\ \text{on } \mathcal{E}^{n} \quad (\text{or } \mathcal{R}^{n})\end{matrix}\right\} \underset{2}{\cong} \left\{\begin{matrix}\text{Versor group} \\ \text{in } \mathcal{R}_{n+1,1}\end{matrix}\right\}$$

Lorentz Trans. $\underline{G}$ (= orthogonal trans.)

$\underline{G}(x) = \; \varepsilon\, G x\, G^{-1} = \sigma x'$ parity $\varepsilon = \pm 1$

$G =$ versor representation of $\underline{G}$ (= spin rep. if $\varepsilon = +1$)

For homogeneous points $x \;\; = x'^2 = 0,$

σ is a scale factor to enforce $e \bullet x' = e \bullet x = 1$

(not a Lorentz invariant)

- Conformal split

  $$G \varepsilon E[\mathbf{x} + \tfrac{1}{2}\mathbf{x}^2 e + e_0]G^{-1} = \; \sigma E\,[\mathbf{x}' + \tfrac{1}{2}(\mathbf{x}')^2 e + e_0]$$

  $\mathbf{x}' = g\,(\mathbf{x})$ is a conformal trans. on $\mathcal{R}^n$

- Versor rep. reduces composition of conformal transformations to versor multiplication

  $$g_3(\mathbf{x}) = g_2[\,g_1(\mathbf{x})\,] \qquad \Longleftrightarrow \qquad G_3 = G_2 G_1$$

- Versor factors $G = s_k \;\ldots\; s_2\, s_1$ vectors $s_i^2 \neq 0$

- Euclidean group $E(n)$ defined by $G\; e\; G^{-1} = \; e$

Rotations & translations generated multiplicatively from reflections

Reflection in the n-(hyper)plane $\qquad n^2 = 1$

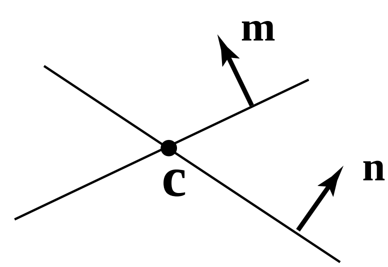
$$\underline{n}(x) = -nxn^{-1} = x' \qquad \sigma = 1$$

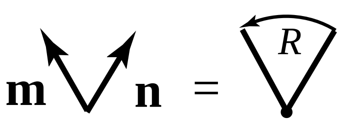
Split form: $\quad n = \mathbf{n}E - e\delta \qquad \mathbf{n}^2 = 1$

For plane thru pt c: $\quad n \bullet c = 0 \quad \Longrightarrow \quad \delta = \mathbf{n} \bullet \mathbf{c}$

- **Rotation** from planes n and m intersecting at point c:

$$G = mn = (\mathbf{m}E + e\mathbf{m} \bullet \mathbf{c})(\mathbf{n}E - e\mathbf{n} \bullet \mathbf{c})$$
$$= \mathbf{mn} - e(\mathbf{m} \wedge \mathbf{n}) \bullet \mathbf{c}$$
$$= R - e(R \times \mathbf{c})$$
$$\mathbf{mn} = R$$



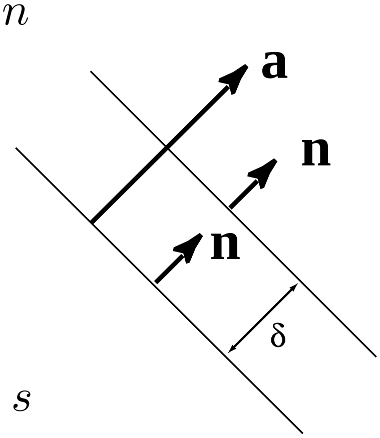


Spinors as directed arcs

- **Translation** from parallel planes m, n

$$G = mn = (\mathbf{m}E - 0)(\mathbf{n}E + e\delta)$$
$$= 1 + \tfrac{1}{2}\mathbf{a}e = T_{\mathbf{a}}$$

where $\quad \mathbf{a} = 2\mathbf{n}\delta$



- **Inversion** generated by sphere vector s

$SE(3) =$ Special Euclidean group on $\mathcal{E}^3$

$= \{$ rigid displacements $\underline{D}\} \underset{2}{\cong} \{$spinors $D\}$

$\underline{D}(x) = Dx\,D^{-1} \qquad D = TR$

Translation by $\mathbf{c}$: $\qquad T = T_{\mathbf{c}} = 1 + \frac{1}{2}e\mathbf{c}$

Rotation axis: $\qquad \mathbf{n} = R\mathbf{n}R^{\dagger}$

- **Chasles' Theorem:** Any rigid displacement can be expressed as a *screw displacement*

Proof: Find a point $\mathbf{b}$ on the screw axis so that

$$D = T_{\mathbf{c}_{\parallel}} T_{\mathbf{c}_{\perp}} R = T_{\mathbf{c}_{\parallel}} R_{\mathbf{b}} = R_{\mathbf{b}} T_{\mathbf{c}_{\parallel}}$$

where

$$R_{\mathbf{b}} = R + e\,\mathbf{b} \times R\,, \quad \mathbf{c}_{\parallel} = (\mathbf{c} \cdot \mathbf{n})$$

Solution: $\qquad \mathbf{b} = \mathbf{c}_{\perp}(1 - R^{-2})^{-1} = \frac{1}{2}\mathbf{c}_{\perp}\dfrac{1 - R^2}{1 - \langle R^2 \rangle}$

- Screw form: $\quad D = e^{\frac{1}{2}S} \qquad S =$ a screw

$se(3) =$ Lie algebra of $SE(3) = \{S = -i\mathbf{n} + e\mathbf{m}\}$

is a bivector algebra,

closed under $\quad S_1 \times S_2 = \frac{1}{2}(S_1 S_2 - S_2 S_1)$

Screw theory follows automatically!

Screws: $$\boxed{S_k = -i\mathbf{m}_k + e\mathbf{n}_k}$$

Product: $S_1S_2 = S_1 \cdot S_2 + S_1 \times S_2 + S_1 \wedge S_2$

Transform: $S_k' = \underline{U}S_k = US_kU^{-1} = Ad_U\, S_k$

$S_1'S_2' = \underline{U}(S_1S_2)$ Product preserving

$= U(S_1 \cdot S_2 + S_1 \times S_2 + S_1 \wedge S_2)U^{-1}$

Invariants: $\underline{U}e = e\,, \quad \underline{U}i = i$

$\Longrightarrow \quad \underline{U}(S_1 \wedge S_2) = S_1 \wedge S_2 = -ie(\mathbf{m}_1 \cdot \mathbf{n}_2 + \mathbf{m}_2 \cdot \mathbf{n}_1)$

$S_1' \cdot S_2' = S_1 \cdot S_2 = -\mathbf{m}_1 \cdot \mathbf{m}_2$ (Killing Form)

Coscrew (Ball's reciprocal screw)

$$\boxed{S_k^* = \langle Sie_0 \rangle_2 = \tfrac{1}{2}(Sie_0 + ie_0S) = i\mathbf{n}_k + \mathbf{m}_k e_0}$$

Invariant: $S_1^* \cdot S_2 = S_2^* \cdot S_1 = \langle\, (S_1 \wedge S_2)\, ie_0 \,\rangle$

$= \mathbf{m}_1 \cdot \mathbf{n}_2 + \mathbf{m}_2 \cdot \mathbf{n}_1$

Pitch: $h = \frac{1}{2}\dfrac{S^* \cdot S}{S \cdot S} = \mathbf{n} \cdot \mathbf{m}^{-1}$

Screw Mechanics (of a rigid body)

I Kinematics of body pt. $x = Dx_0 D^{-1}$ $\quad D = D(t)$

$$\dot{D} = \tfrac{1}{2} V D \qquad \dot{x} = V \bullet x$$

- $V = -i\boldsymbol{\omega} + e\mathbf{v} \quad \Longrightarrow \quad \dot{\mathbf{x}} = \boldsymbol{\omega} \times \mathbf{x} + \mathbf{v}$

  = "instantaneous screw" $\quad \mathbf{v} =$ CM velocity

- $P = \underline{M}\, V = i\underline{I}\boldsymbol{\omega} + m\mathbf{v}e \;\; = i\ell + \mathbf{p}\, e$

  = comomentum (a coscrew)

II Dynamics

- $\dot{P} = W = i\Gamma + \mathbf{f}\, e_0 =$ coforce (wrench)

$$\Longrightarrow \quad \dot{K} = V \bullet W = \boldsymbol{\omega} \bullet \Gamma + \mathbf{v} \bullet \mathbf{f} = \text{Power}$$

$$K = \tfrac{1}{2} V \bullet P = \tfrac{1}{2} \boldsymbol{\omega} \bullet \ell + \mathbf{v} \bullet \mathbf{p} = \text{K.E.}$$

More in NFCM

III Change of Frame $\quad x \longrightarrow x' = \underline{U} x = U x U^{-1}$

$$\Longrightarrow \quad V' = \underline{U}\, V \qquad \textit{Covariant} \qquad \dot{U} = 0$$

$$P = \overline{U} P' \qquad \textit{Contravariant}$$

$$P' \bullet V' = P \bullet V \qquad \textit{Invariant}$$